

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

SIP KLIENT S POKROČILÝMI FUNKCEMI

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

VOJTĚCH ŠACL

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

SIP KLIENT S POKROČILÝMI FUNKCEMI

SIP CLIENT WITH ADVANCED FEATURES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

VOJTĚCH ŠACL

VEDOUcí PRÁCE
SUPERVISOR

Ing. MARTIN ZUKAL

BRNO 2013



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Vojtěch Šacl

ID: 134624

Ročník: 3

Akademický rok: 2012/2013

NÁZEV TÉMATU:

SIP klient s pokročilými funkcemi

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a zrealizujte vlastní řešení, které umožní uskutečnit hlasový i video (V2oIP) hovor z webového prohlížeče bez nutnosti instalace jakéhokoli dalšího softwaru na koncový počítač. Jako signalizační protokol využijte protokol SIP. Vámi vytvořené řešení bude také umět zasílat a přijímat rychlé zprávy (instant message). Aplikace bude spolupracovat s některým ze zdarma dostupných SIP serverů. Dále bude aplikace obsahovat seznam kontaktů, které budou získány ze SIP serveru. Funkčnost aplikace ověřte testováním.

DOPORUČENÁ LITERATURA:

- [1] JOHNSTON, A. B. SIP: Understanding the Session Initiation Protocol, Second Edition. London: Artech House Publishers, 2004. ISBN 978-1580536554
- [2] FETTE, I., MELNIKOV, A. The WebSocket Protocol [online]. December 2011, [cit. 2012-10-09]. Dostupný z WWW: <<http://tools.ietf.org/html/rfc6455>>
- [3] CASTILLO, I. B., VILLEGAS, J.M., PASCUAL V. The WebSocket Protocol as a Transport for the Session Initiation Protocol (SIP) [online]. September 7 2012, [cit. 2012-10-09]. Dostupný z WWW: <<http://tools.ietf.org/html/draft-ietf-sipcore-sip-websocket-03>>

Termín zadání: 11.2.2013

Termín odevzdání: 5.6.2013

Vedoucí práce: Ing. Martin Zukal

Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

PODĚKOVÁNÍ

Výzkum popsáný v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....
(podpis autora)

ABSTRAKT

Hlavním cílem této bakalářské práce je návrh a realizace softwarového klienta, který bude schopen provádět textovou i video komunikaci přes síť založenou na protokolu IP. Tato aplikace by měla být spouštěna přímo z webové stránky a neměla by vyžadovat instalaci žádného dalšího software na koncový počítač. V první části jsou popsány možné technologie pro vytvoření této aplikace. Dále následuje stručný popis použitých technologií. Důraz je kladen především na protokol SIP. V práci je stručně popsána charakteristika několika knihoven, které byly uvažovány pro implementaci do aplikace. Návrh řešení obsahuje popis grafického prostředí a chování aplikace. V poslední části práce je výsledná aplikace otestována v různých modelových situacích.

KLÍČOVÁ SLOVA

AJAX, JavaScript, klient, PHP, programování, SIP, SIP server, SDP, VoIP, WebRTC, WebSocket, WebSocket SIP podprotokol

ABSTRACT

The main objective of this thesis is the design and implementation of client software, which will be able to conduct text and video communication over an IP based network. The client application is supposed to run from a web page without the need of any additional software. The first part describes possible technologies suitable for the development of such an application. After that follows a brief description of the selected technology with emphasis on SIP protocol. The thesis also includes a brief description of characteristics of libraries that were considered for implementation into the application. The proposed solution consists of the description of the graphic environment and behavior of the application. The developed application is tested in different situations in the last part of the thesis.

KEYWORDS

AJAX, JavaScript, client, PHP, programming, SIP, SIP server, SDP, VoIP, WebRTC, WebSocket, Sec-WebSocket-Protocol

ŠACL, Vojtěch *SIP klient s pokročilými funkcemi*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. 61 s. Vedoucí práce byl Ing. Martin Zukal

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „SIP klient s pokročilými funkcemi“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu semestrálního projektu panu Ing. Martinu Zukalovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

(podpis autora)

OBSAH

Úvod	11
1 Technologie pro realizaci hovorů z prohlížeče	12
1.1 Java Applet	12
1.2 Alternativy	13
1.3 Volba technologií	14
2 Teoretický popis použitých technologií	16
2.1 JavaScript	16
2.2 PHP	18
2.3 AJAX	19
2.4 WebRTC technologie	20
2.4.1 Architektura	21
2.5 SIP Protokol	23
2.5.1 SIP síťové prvky:	23
2.5.2 Princip komunikace	25
2.5.3 Jiné typy zpráv	28
2.5.4 Odpovědi	29
2.5.5 Protokol SDP	32
2.6 WebSocket protokol	33
2.6.1 WebSocket SIP podprotokol	34
3 Návrh vlastního řešení	37
3.1 Cíl práce	37
3.2 Knihovny	37
3.2.1 Flashphoner	38
3.2.2 sipML 5	38
3.2.3 JsSIP	38
3.2.4 Výběr dalších knihoven	39
3.3 Grafické prostředí	39
3.4 Architektura	41
3.5 Popis jednotlivých funkcí	43
3.5.1 Registrace k serveru	43
3.5.2 Přihlášení k serveru	43
3.5.3 Funkce k obsluze videohovorů	45
3.5.4 Zprostředkování rychlých zpráv	45

4	Testy aplikace	48
4.1	Přihlášení klienta k SIP serveru přes prohlížeč Chrome	48
4.2	Přihlášení klienta k SIP serveru přes prohlížeč Firefox Nightly	51
4.3	Videohovor uživatelů v prohlížeči Chrome	51
4.4	Videohovor uživatelů v prohlížeči Ff Nightly	53
4.5	Videohovor uživatelů mezi prohlížeči Chrome a Firefox Nightly	53
4.6	Konverzace dvou uživatelů mezi prohlížeči Chrome a Firefox Nightly	54
4.7	Konverzace tří uživatelů v prohlížečích Chrome	55
5	Závěr	56
	Literatura	57
	Seznam symbolů, veličin a zkratk	59
A	OBSAH CD	61

SEZNAM OBRÁZKŮ

2.1	Přenos JavaScript kódu	16
2.2	Princip AJAX technologie	19
2.3	Architektura technologie WebRTC	21
2.4	Výměna zpráv SIP protokolu	25
2.5	Registrace uživatelů přes WebSocket protokol.	35
3.1	Grafické prostředí SIP klienta	40
3.2	Architektura videohovoru	41
3.3	Architektura rychlých zpráv	42
4.1	Vyobrazení webové stránky v Chrome a konzole SIP serveru	49
4.2	Vyobrazení webové stránky Firefox Nightly a konzole SIP serveru	51
4.3	Videohovor dvou vzdálených uživatelů v prohlížeči Chrome	52
4.4	Konverzace dvou uživatelů v prohlížečích Firefox Nightly a Chrome	54
4.5	Konverzace tří uživatelů v prohlížeči Chrome	55

SEZNAM TABULEK

2.1	Další významné metody	29
2.2	Odpovědi 1xx	29
2.3	Odpovědi 2xx	30
2.4	Odpovědi 3xx	30
2.5	Odpovědi 4xx	31
2.6	Odpovědi 5xx	31
2.7	Odpovědi 6xx	32
2.8	Rezervované rozsahy Status kódů	33
2.9	Definované Status kódy WebSocket protokolu pro uzavření rámce . .	34

ÚVOD

Přenos hlasu přes internetové připojení nebo-li VoIP (Voice over Internet Protocol) se dnes značně rozšiřuje. Jeho rozvoj je datován od posledních let 20. století. Výhodou je cena hovoru, většinou se platí jenom za datové připojení. Jeho nejzákladnější sestava se skládá ze dvou VoIP zařízení a spojovacího média. Pro jeho alespoň dostatečnou realizaci je zapotřebí mít zajištění tzv. kvality služby QoS (Quality of Service) [1]. Ta určuje různé parametry, od zajištění dostatečné přenosové kapacity, zpoždění přenosu až po ztrátovost paketů.

Signalizace pro přenos dat je řešena různě, závisí na použité architektuře.

Nejstarší architekturou pro přenos hlasu na internetu je doporučení Mezinárodní Telekomunikační unie ITU-T standard H.323. Jedná se o soubor protokolů, které podporují kodeky a vybudování celé relace.

Jednou z možných alternativ pro VoIP je IAX (Internet Asterisk eXchange) protokol [2]. Jedná se o open source protokol, což značí otevřený veřejně publikovaný zdrojový kód. Rozdílovým faktorem je, že signalizační a multimediální data se přenáší pouze jedním datovým tokem na jednom portu. Je použit i ve vlastní pobočkové ústředně společnosti Asterisk.

Extensible Messaging and Presence Protocol (XMPP) [3] je open source protokol (obdoba SIP protokolu), přes který je realizován nejenom chat (popř. rychlé zprávy). Vylepšení Jingle využívá RTP, proto je kompatibilní s protokoly SIP a SDP. Toto řešení využívá Google Talk.

Dnes se stává populární protokol Session Initiation Protocol – SIP. Samotný SIP je pouze signalizační protokol, který využívá další protokoly. Pro přenos multimediálních dat využívá Real-time Transport Protocol (RTP) a pro popis přenosu Session Description Protocol (SDP). Výhodou SIP protokolu je snadný přechod přes místa překládající síťové adresy NAT (Network Address Translation) a spojení za ním.

Cílem práce je navrhnout softwarového klienta, který bude schopen sestavit hovor pomocí signalizačního SIP protokolu. Pro uživatele nabídne pokročilé funkce. Tím se rozumí video chat (přenos video obrazu v reálném čase oběma klienty zároveň). Dále navrhovaný SIP klient umožní posílat tzv. rychlé zprávy (instant messaging) uživatelům registrovaným k vlastnímu SIP serveru. Softwarový klient bude umístěn ve webové stránce, která bude soběstačná. Tím se rozumí, že nebude k bezproblémovému chodu zapotřebí instalace dalšího softwaru na koncový počítač. Vše by měly obstarat technologie obsažené v prohlížeči - WebSocket a WebRTC.

1 TECHNOLOGIE PRO REALIZACI HOVORŮ Z PROHLÍŽEČE

Na trhu desktopových prohlížečů můžeme říci, že existuje pět hlavních společností. Pro operační systém od Microsoftu MS Windows se uplatňuje vlastní Internet Explorer. V poslední době se rozmáhá obliba uživatelsky jednoduchého prohlížeče Chrome od společnosti Google. Dalšími zástupci jsou prohlížeč Firefox od společnosti Mozilla, prohlížeč Opera od Opera Software a Safari od společnosti Apple.

Do prohlížečů je již implementován nový standard HTML ve verzi pět [4], který je ve stádiu návrhu konsorciem W3C (World Wide Web Consortium). Verze pět oproti svým starším verzím přehraje multimediální obsah (video, audio) přímo ve webovém prohlížeči bez potřeby využití různých zásuvných modulů (Flash).

Internetové servery nabízejí audio (popř. video s audiem) telefonní klienty zabudované ve webovém prohlížeči.

Zde pár příkladů:

- Eyejot ¹ podporuje tvorbu a přenos video e-mailů. K plně funkční aplikaci je nutné připojení k internetu, headset (sluchátka s mikrofonom, popř. i webkamera) a nainstalovaný zásuvný modul Flash.
- Společnosti C2Call ² a DoddlePhone ³ umožňují internetové volání z webového prohlížeče. V prohlížeči je nutné mít nainstalovaný zásuvný modul jazyku Java.

1.1 Java Applet

Programovací jazyk Java je objektově orientovaný jazyk vytvořen společností Sun Microsystems. Java se řadí mezi nejpoužívanější programovací jazyky na světě. Díky své přenositelnosti je používán v programech, které pracují na různých systémech. V praxi se jazyk Java vyskytuje na čipových kartách (JavaCard), v mobilních telefonech (JavaME) nebo v aplikacích umístěných na webových stránkách. O správnou funkci Java programů na různých platformách (Windows, Linux, Mac OS) se stará virtuální stroj jazyka Java, zkráceně JVM (Java Virtual Machine), který je pro každou platformu jiný, ale dostupný pro výše zmíněné.

Applet (program v jazyce Java) se při otevření webové stránky stáhne do paměti počítače a obsažený bajtový kód je spuštěn přes JVM. Webové prohlížeče mohou

¹Společnost EyEjot je k nalezení na <http://www.corp.eyejot.com/>

²Webová adresa společnosti C2Call je <http://www.c2call.com/>

³Webová adresa společnosti DoddlePhone je <http://www.doddlephone.com/>

vyžadovat instalaci JVM z důvodu, že zásuvný modul není k dispozici a je potřeba provést instalaci samostatně. Applet může mít zpřístupněny některé vlastnosti a metody jako veřejné, což umožňuje například technologii JavaScript definovat chování a propojovat Java applet s různými částmi webové HTML stránky.

Java applety jsou využívány ke grafickým efektům pro oživení stránek. Jejich nevýhodou oproti dynamickému HTML a JavaScript je pomalá rychlost vzniklá nutností načítat velké množství tříd. Java applet lze často nalézt tam, kde skriptovací jazyky nestačí – složitější algoritmy.

Základní prvek Java appletu tvoří třída `java.applet.Applet`, která definuje základní metody rozhraní mezi prohlížečem a appletem. Podrobnější popis Java appletu lze nalézt v [5].

1.2 Alternativy

ActiveX

ActiveX [6] je softwarová komponenta vyvinutá společností Microsoft a je omezena funkčností pouze na operačním systému MS Windows. Podstata je v tom, že každá nová aplikace nemusí být programována od začátku. Mohou být použity tzv. ActiveX controls, což jsou malé programy, které obsahují určitou funkci. Společnost VaxSoft navrhla vlastní SIP klient s instrukční sadou zahrnující ActiveX technologii (knihovny).

Silverlight

Technologie byla vyvinuta pro tvorbu a prohlížení dynamického online obsahu. Při stáhnutí zásuvného modulu umožňuje ovládání her nebo přehrávání multimediálního obsahu. Je dostupný pro většinu prohlížečů s operačním systémem MS Windows nebo Mac OS X. Byl vyvinut společností Microsoft a sdílí velmi podobné vlastnosti jako Flash. Byl vyvinut Silverlight SIP klient společností Ozeki Systems Ltd., která nabízí výběr mezi technologiemi Flash a Silverlight.

ASP.NET

SIP.NET klient byl vyvinut technologií ASP.NET, která je prvkem .NET Framework. Je podporován operačním systémem Windows a je napsán v jazyce C#⁴.

⁴Pro inspiraci je uvedena aplikace od společnosti Independentsoft, jenž je napsána v jazyce C#: <http://www.independentsoft.de/sip/tutorial/registerasp.html>.

Mezi prvními přišli vývojáři OverSIP a JsSIP knihovny s nápadem vytvořit SIP klienta nezávislého na instalaci zásuvných modulů, pouze s protokoly HTML 5, WebSocket a technologií WebRTC. Společnost Google má velký podíl na vývoji WebRTC a vydala vlastní WebRTC API jako otevřený zdrojový kód pro další vývoj. Společnost Doubango Telecom použila tyto knihovny od Google a vytvořila vlastní aplikaci používající i HTML 5 a JavaScript.

1.3 Volba technologií

Základním parametrem výběru správné technologie byla podmínka funkce ve webovém prohlížeči, který byl nainstalován na koncovém počítači.

U zásuvných modulů ve webových prohlížečích není jejich stabilita na vysoké úrovni. Také výkon a uzavřenost celého systému nepřidá na hodnocení. Flash player často stojí za pádem webového prohlížeče.

Flash aplikace jsou velmi rozšířené, navíc podporují kodeky, které jsou uzpůsobené pro VoIP. Mezi hlavní nevýhody se řadí bezpečnost. Společnost Mozilla proto ve svém prohlížeči zobrazuje varování. Společnost Google dokonce Flash integrovala trvale do svého prohlížeče, aby mohla aplikovat vlastní bezpečnostní technologii. Společnost Adobe přestala vyvíjet technologii Flash ⁵, bude pouze vydávat kritické záplaty chyb bezpečnostní aktualizace. Ve skutečnosti už jsou Flash aplikace na ústupu, jelikož jejich náhradou se stává standard HTML 5. S novým standardem HTML 5 je možné pracovat pohodlněji s multimediálním obsahem.

Výhodou technologie Java je bezpečnost. Speciálním typem Java programu je applet, který kvůli své bezpečnosti pracuje pouze v rámci tzv. bezpečnostního pískoviště (sandbox), kde má velmi omezený přístup k prostředkům. Java applet nesplňuje podmínku vytvoření SIP klienta bez přidáných zásuvných modulů. Více informací naleznete v kapitole 1.1.

Z předešlých odstavců vyplývá, že pro vývoj SIP klienta použijeme technologie aplikované přímo v prohlížeči. Tyto technologie se jmenují WebSocket a WebRTC, a zaručí nám plně funkční aplikaci ve webovém prohlížeči. Tyto technologie jsou zabudovány ve vlastním webovém prohlížeči a podporují přenos multimediálních dat mezi uživateli.

WebSocket protokol vytváří spolehlivé trvalé spojení mezi klientem a serverem, které je udržováno pro komunikaci v reálném čase. Pomocí SIP protokolu můžeme

⁵Společnost Adobe se začala zaměřovat na vyvíjející se standard HTML 5, což oznamuje v tiskové zprávě: <http://blogs.adobe.com/conversations/2011/11/flash-focus.html>

navázat klasický telefonní hovor využívající jeho signalizační metody a odpovědi.

WebRTC je poměrně novým prvkem v internetovém světě, proto jeho hranice využití nejsou konečné. I když už je implementován ve webovém prohlížeči, je stále ve vývoji a v procesu schvalování. Tím webovým prohlížečem se stal Chrome od Google, kde jeho starší verze označením 21 a 22 podporovaly tuto technologii v testovacím režimu. Ve verzi 23 je technologie plně dostupná. Také společnost Mozilla plánuje integrovat tuto technologii do svého webového prohlížeče Firefox ⁶. Implementace WebRTC do webových stránek v jazyce JavaScript je jednoduchá a spojení s webovou kamerou (popř. mikrofonom) je otázkou pár řádků kódu.

Většina webových stránek se píše v jazyce JavaScript, které vytvoří stránku vzhledově i obsahově zajímavou s funkčními prvky. Tyto prvky mají své vlastnosti, příkazy a hodnoty zabudované v kódu. Ty se stahují s celou stránkou, což ušetří zatížení serverům a nechává výpočetní výkon na prohlížečích uživatelů.

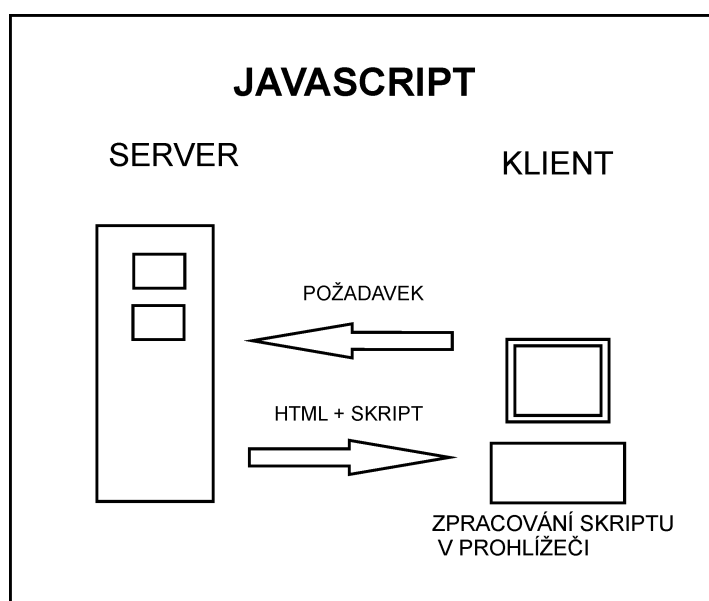
⁶Mozilla oficiálně nevyjádřila datum vydání a verzi prohlížeče, který bude podporovat technologii WebRTC, neoficiálně se ale mluví o datu leden roku 2013 a verzi 18. Webová adresa společnosti Mozilla zabývající se problematikou WebRTC: <https://developer.mozilla.org/cs/docs/WebRTC>.

2 TEORETICKÝ POPIS POUŽITÝCH TECHNOLOGIÍ

2.1 JavaScript

Jazyk JavaScript byl vyvinut ve společnosti Netscape v roce 1995. Je to kompaktní skriptovací jazyk, který je vložen do webových stránek pro jejich oživení. JavaScript je přítomen téměř kdekoli, představuje totiž jazyk webu. JavaScript přímo nesouvisí s Javou. Když byl JavaScript poprvé představen, jazyk Java byl dominantní. Nicméně prosazení jazyku JavaScript v prostředí webu bylo razantní a uživateli byl velice kladně přijat. Určitou výhodou je, že JavaScript zůstal časem poměrně nezměněn a zakládá si na své jednoduchosti i rychlosti. Postupem času se Java stala relativně složitý a spleť jazyk.

JavaScript je vložen v zdrojovém kódu webové stránky a odeslán ze serveru ke klientovi, jehož prohlížeč vložený skript provede.



Obr. 2.1: Přenos JavaScript kódu

Jednoduchý skript v jazyku JavaScript vložený v těle HTML stránky vypadá následovně:

```
<body>
<p>Toto je normální text stránky. </p>
<script language="JavaScript">
```

```
document.write("Ahoj světe");  
</script>  
</body>
```

Výpis některých vlastností a výhod je uveden dále [7].

- Jazyk JavaScript podporuje nepřeberné množství efektních i efektivních možností pro webové stránky. Umožní ovládat všechny prvky na webové stránce i samotnou webovou stránku.
- Dynamické HTML (DHTML) je souhrnným názvem pro celý soubor postupů. Od zpracování obrázku, přes libovolné umístění prvků na stránku, až po použití speciálních objektů DHTML pro optické efekty.
- Zprvu byly webové stránky tvořeny způsobem, že zasílaly data zpět na server kvůli ověření jejich správnosti. Tak byla vytvořena časová ztráta, navíc při pomalém připojení se ztráta značně zvětšila. Vstupní data, které uživatel umístil na webovou stránku, dokáže ověřit JavaScript pomocí skriptu ještě před jejich odesláním zpět na server. Princip komunikace mezi serverem a klientem je zachycen na obrázku 2.1.
Například: Zadal uživatel telefonní číslo v platném formátu? Je správně nastavena doručovací adresa?
- Zpracování cookies patří k populárnímu využití jazyku JavaScript. Cookies jsou malé soubory, které si pamatují různé údaje. Webový prohlížeč si dokáže pamatovat hesla, uživatelská jména nebo dokonce celé nastavení stránky.
- Jazyk JavaScript umožňuje práci se soubory XML a XLST, také používá šablony stylů CSS.
- Je dobré psát skripty až na konec stránky. Umožní to rychlejší načítání webových stránek pro uživatele, který má vypnutý JavaScript. Není to ale povinné.

Bližší informace o jazyku JavaScript naleznete na [7].

2.2 PHP

Jazyk PHP (Hypertext Preprocessor) patří mezi nejrozšířenější technologie vývoje webových aplikací. PHP vzniklo v roce 1996 a bylo původně pojmenováno Personal Home Page. Technologie PHP spadá do kategorie Otevřený zdrojový kód (možnost využívat zdarma) a existuje velká podpora v oblasti hostování webových aplikací.

Nejnovější verze jazyka se označuje PHP 5. Mezi hlavními novinkami verze je podpora objektově orientovaného programování a podpora jazyka XML, proto se může přizpůsobit současným technologiím a trendům.

Jazyk PHP pracuje na straně serveru [8]. Data webových stránek jsou uložena na webovém serveru. Nejprve se PHP stránka provede na hostujícím serveru a potom se výsledný HTML kód odešle webovému prohlížeči. Výhodou tohoto přístupu je, že se ke zdrojovým kódům nedostane nikdo, kdo nemá přístup k serveru, na kterém jsou soubory umístěny. Samotné PHP skripty se mohou zapisovat přímo do běžné HTML stránky, pouze se musí souboru přejmenovat přípona na .php, pro zpracování u serveru.

Díky jazyku PHP je možnost vytvořit mnoho aplikací, např. knihu návštěv, diskuzní fórum, anketu, počítadlo. Navíc se nabízí možnost propojit webové stránky s databází, např. MySQL, PostgreSQL nebo dalšími.

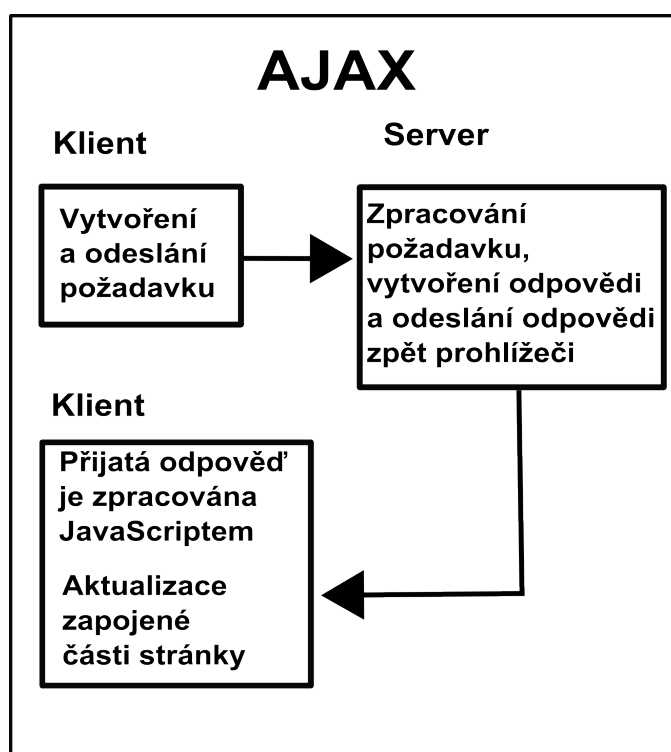
Vytvoření jednoduchého skriptu:

```
<?php
$retez = "Ahoj";
echo $retez;
?>
```

Zápis skriptu v PHP jazyce se uzavírá mezi párové značky `<?php` a `?>`. Typ proměnné není nutné definovat, sám se určí po přiřazení hodnoty. Název proměnné začíná znakem `$` a bez mezery následuje pojmenováním. První znak musí být písmeno abecedy (a-Z), tudíž ne číslo nebo speciální znaky. Názvy proměnných rozlišují malá a velká písmena. Pole může obsahovat ve svých prvcích různé datové typy.

2.3 AJAX

Technologie AJAX [9] (Asynchronous JavaScript and XML) umožňuje, aby webová stránka odeslala požadavek na server a obdržela od něj libovolná data v XML. Princip funkce technologie AJAX je zobrazen na obrázku 2.2. AJAX využívá současné technologie k vytvoření dynamičtějších webových stránek. Využívá se jazyku JavaScript, který způsobí, že stránka nemusí být načtena v plném rozsahu, ale pouze nezbytná data, která se na webové stránce mění. Čímž se snižuje doba prodlení, kde uživatel čeká na odezvu systému. Protože přenos dat se děje na pozadí, může uživatel nerušeně pracovat v systému.



Obr. 2.2: Princip AJAX technologie

Průběh celé relace popíšeme ve třech částech:

1. Nejprve se na straně klienta vytvoří požadavek - Objekt typu XMLHttpRequest. Tato komunikace spočívá ve vytvoření HTTP požadavku na soubor nebo skript umístěn na serveru. Objekt XMLHttpRequest může být odeslán buď metodou GET nebo POST. Do objektu XMLHttpRequest lze vložit vstupní data, se kterými pak server pracuje. Tyto požadavky jsou jednoduché a nezpůsobují problémy s firewallem.

2. Po odeslání se čeká na reakci a odpověď serveru. Je nutné, aby server vlastnil potřebné technologie pro zpracování požadavku. Po vykonání serverového skriptu se zašle odpověď pomocí HTTP. Doporučeným formátem odpovědi je XML, jehož výhodou je velké rozšíření a značné množství knihoven. Samozřejmě je možné použít i jiné formáty, které dokáže JavaScript zpracovat.
3. Následuje samotné zpracování odpovědi, kde JavaScript vykonává požadované úkony na webové stránce bez znovunačtení celé stránky. Aktualizuje se pouze změněná část stránky.

Jednoduchá AJAX funkce

```
$.ajax({  
  url : "main.php",  
  type : "GET",  
  data : {data2 : login},  
  success : function(xml) {$("#poleID").text();}  
});
```

Pro použití technologie AJAX byla použita knihovna jQuery, která velmi zjednodušuje zápis celé funkce. V řádku `url` byl zavolán soubor `main.php`, který bude vykonán. Pomocí metody `GET` byla předána vstupní data z JavaScript proměnné `login` a uložena do proměnné `data2`, která je načtena v souboru PHP. Po úspěšném dokončení jsou výstupní data vypsána do HTML stránky.

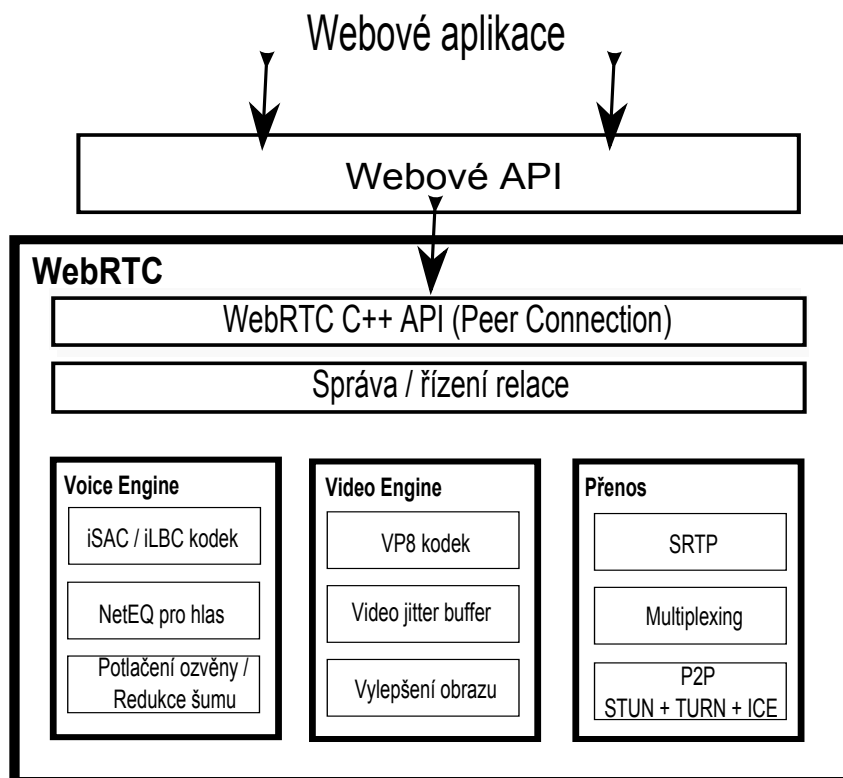
Jelikož při práci s AJAXem se nemění URL adresy stránek, není možnost si adresu uložit do záložek, aniž bychom nepřišli o vykonanou AJAX funkci. Pokud má uživatel vypnutou podporu jazyku Javascript, není možnost splnit AJAX funkce.

2.4 WebRTC technologie

WebRTC je open source projekt pracující s multimediálním obsahem, který je odeslán i přijímán v reálném čase. Technologie WebRTC je implementovaná v prohlížeči Chrome a ve vývojové verzi prohlížeče Firefox. Byla vyvinuta společností Google, která má hlavní podíl na jeho rozvoji, ale je dostupná i pro webové prohlížeče Firefox a Opera, jejichž autoři se také podílejí na vývoji. (Podpora od Microsoftu je sporná, ten čeká, až bude WebRTC schválen standardem). Technologie WebRTC byla pouze definovaná konsorciem W3C [10].

Využití nalezne nejenom u internetového volání, video chatu a u sdílení souborů. WebRTC využívá vaši webkameru a mikrofon pro pořízení multimediálního obsahu.

Například obraz z webkamery se zobrazuje v okně prohlížeče. Toto bylo umožněno ještě před technologií WebRTC, ale jen za pomoci zásuvných modulů. WebRTC využívá video kodek VP8, jenž je uzpůsoben pro video chat. Avšak není podporován většinou ostatních SIP klientů (hardwarové nebo softwarové, které používají H.264), což je nevýhoda a je potřeba použít překladače.



Obr. 2.3: Architektura technologie WebRTC

2.4.1 Architektura

Webová aplikace

Aplikace v prohlížeči, které využívají technologii WebRTC pro multimediální komunikaci.

Webové API

WebRTC API obsahuje sadu instrukcí (příkazů), které využívají možnosti technologie WebRTC. Tyto příkazy jsou psány v jazyce JavaScript a jsou součástí zdrojového kódu. Například metoda `getUserMedia` technologie WebRTC vyvolá požadavek k přístupu na webovou kameru. Více informací naleznete na [10] a [11].

WebRTC C++ API

Část API, která se zaměřuje na odeslání nebo propojení s druhým klientem pro výměnu multimediálních dat z vlastního webového prohlížeče. Více informací naleznete na [12].

Správa relace

Abstraktní relační vrstva umožňující nastavení a řízení hovorů. Ponechává implementaci protokolu na vývojáři aplikace.

Přenos

Obsahuje tři stěžejní funkce, které zajišťují bezprostřední komunikaci pro protokol WebRTC. Pro komunikaci je používán zabezpečený Real-Time Protokol (SRTP). Dále využívá možnosti multiplexování, to znamená slučování více signálů do jednoho datového toku. Také obsahuje mechanismy pro vytvoření spojení napříč různými typy sítí.

VoiceEngine

Technologie pro nepřetržité pořizování audio signálu, kde data jsou přenášena ze zvukové karty do sítě. Obsahuje hlasové kodeky pro VoIP hovor a kódování audia. Pro zlepšení kvality využívá algoritmus pro minimalizaci ozvěny v hlase a redukci šumu z okolí.

VideoEngine

Technologie pro nepřetržité pořizování obrazového materiálu pocházejícího z kamery (popř. fotoaparátu) určeného pro přenos do sítě nebo na obrazovku. Obsahuje video kodek VP8 z WebM Project ¹, který se vyznačuje nízkým zpožděním. Pro zlepšení kvality jsou k dispozici funkce, které zakryjí účinky chvění, ztrátu paketů a odstraňují obrazový šum.

¹Internetová stránka objasňující kodek VP8: <http://www.webmproject.org/docs/vp8-sdk/index.html>

2.5 SIP Protokol

Protokol SIP se začal vyvíjet v roce 1996 skupinou MMUSIC (Multiparty Multimedia Session Control) v rámci IETF (Internet Engineering Task Force). V roce 1999 vyšla specifikace standardu jádra protokolu RFC 2543 a zároveň byla založena pracovní skupina s názvem SIP. V roce 2002 byl vydán nový standard RFC 3261 [13], který nahradil původní. K účelu vylepšení protokolu SIP vznikly ještě jiné pracovní skupiny, např. SIPPING (zabývající se aplikacemi a návrhy dalšího vývoje SIP), SIMPLE (služby zjišťování přítomnosti a předávání zpráv), IPTEL (číslování a směrování). Protokol SIP je textově orientovaný, pojetí má podobné jako HTTP protokol.

Je to signalizační protokol na aplikační vrstvě závisící na protokolech nižších vrstev. Protokol SIP neurčuje, jaký bude použit transportní protokol, jak bude řešen přenos multimediálních dat, řízení hovoru. Zato řeší sestavení spojení dvou a více účastníků a jeho dohled. Transportním protokolem je nejčastěji UDP (User Datagram Protocol), který sice není spolehlivý (není zaručeno 100% přijetí paketů), ale má menší zpoždění oproti TCP. Pro multimediální zprávy se používá RTP standardizovaný v RFC 3550 [14].

U protokolu SIP není podstatná až tak spolehlivost přenosu, jako rychlost a jednoduchost přenosu. Uvnitř zprávy protokolu SIP je totiž zapouzdřena zpráva jiného protokolu, ať se již jedná o kódování audio dat, video dat, parametry a porty, na kterých vysílá nebo přijímá. Pro tento účel existuje textový protokol SDP, který popisuje inicializační parametry odesílaného média.

2.5.1 SIP síťové prvky:

- Proxy server
- Registrar server
- Location server
- Redirect server
- User agent

Proxy server

Přijímá žádosti od uživatele nebo od jiného Proxy serveru. Pokud nezná volaného, předává žádost jinému Proxy serveru, dokud některý neodpoví. Poté volanému posílá žádost o spojení.

Proxy servery se dělí:

- **Stateful** (informace o stavech), po přijetí požadavku vytvoří stav a ten drží, dokud spojení není ukončeno
- **Stateless** (bezstavový) přeposílají zprávy nezávisle na vzájemných vazbách

Registrar server (Registrační server)

Přijímá od uživatelů požadavky na registraci, tím získává informaci o jejich umístění (uživatelské jméno, IP adresa a port) a předává ji Lokačnímu serveru. Klient si registruje účet, přitom specifikuje dobu, po kterou chce být k serveru registrován. Pokud hodnotu nespecifikuje, platí standardní jedna hodina. Registraci je nutné obnovovat.

Location server (Lokační server)

Lokační server vlastní informace o umístění uživatelů pro Proxy server a Server přesměrování. Komunikace mezi Lokačním serverem a ostatními neprobíhá SIP protokolem, ale nejčastěji pomocí LDAP protokolu (Lightweight Directory Access Protocol), který přistupuje k adresářovým službám online. V praxi jsou tyto servery zakomponovány v sobě, a chovají se buď jako Proxy server nebo Server přesměrování.

Redirect server (Server přesměrování)

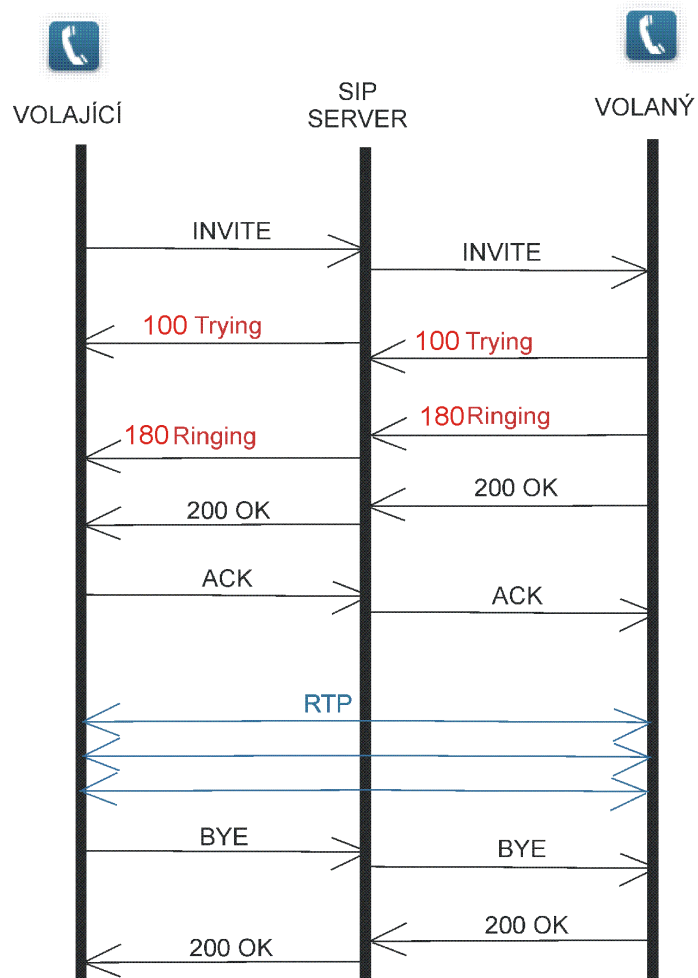
Server přesměrování pracuje obdobně jako Proxy server, jenom nepřeposílá žádosti. Pouze sdělí uživateli adresu serveru volaného uživatele a ten se spojí sám s druhou stranou. Adresu zjišťuje pomocí lokalizační služby. Když je lokalizace neúspěšná, server nedisponuje žádnou adresou. Naopak je-li k dispozici více adres, může klient kontaktovat více serverů.

User Agent (Uživatel)

Koncový uživatel SIP sítě. Nejčastěji jsou reprezentovány IP telefony, které rozdělujeme na fyzické nebo softwarové instalované do počítače. Uživatel se skládá ze dvou částí pracujících současně, a spojení může ukončit každý samostatně.

- **Uživatel Klient:** má na starost inicializaci spojení .
- **Uživatel Server:** reaguje na příchozí žádosti a odesílá odpovědi.

2.5.2 Princip komunikace



Obr. 2.4: Výměna zpráv SIP protokolu

Na začátku celého spojení se použije zpráva typu INVITE:

```
INVITE sip:13@10.10.1.13 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.99:5060;branch=z9hG4bK343bf628
Max-Forwards: 70
From: "Test 15" <sip:15@10.10.1.99>;tag=as58f4201b
To: <sip:13@10.10.1.13>;tag=bf48a32hdk
Contact: <sip:15@10.10.1.99>
Call-ID: 326371826c80e17e6cf6c29861eb2933@10.10.1.99
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 258
```

```
v=0
o=root 1821 1821 IN IP4 10.10.1.99
s=session
c=IN IP4 10.10.1.99
t=0 0
m=audio 11424 RTP/AVP 0 8 101
a=rtpmap:0 PCMU/8000
```

První řádek SIP zprávy značí, že se jedná o žádost typu INVITE, která se používá pro navázání spojení. Dále je zmíněn koncový uživatel spojení a verze protokolu SIP. Adresy uživatelů spojení jsou podobné emailovým adresám (pouze slovní názvy jsou nahrazeny IP adresami pro kratší zápis), mají i podobnou funkci a pro každého uživatele jsou individuální pro jednoduchou identifikaci na globální úrovni. To znamená, že se uživatel může připojit odkudkoliv a přihlásit se prostřednictvím svého SIP URI (Uniform Resource Identifier).

Pole **Via** je používané v žádostech i odpovědích a obsahuje jméno a verzi protokolu SIP, transportní protokol (v tomto případě UDP) a cestu žádosti, která by měla být dodržena při poslání odpovědi. Uzel je reprezentován klientem nebo serverem. Pole **Via** zakončuje parametr **branch**, který je pro potřeby identifikace. V každé odpovědi na jednu žádost je parametr **branch** stejný. Hodnota **branch** vždy začíná kódem: **z9hG4bK**. Zpráva může obsahovat více polí **Via**, kdy server, přes který zpráva prochází, vždy jedno přidá.

Pole **Max-Forwards** udává hodnotu maximálního počtu Proxy serverů a bran, které byly využity po cestě. Každý uzel odečte 1 z příchozí hodnoty. Podobným způsobem pracuje i parametr Time To Live (TTL) z protokolu IP.

Ve zprávě nesmí chybět pole **From** a **To**, které obsahují SIP adresy spojovaných uživatelů. Parametr **tag** je náhodný řetězec generovaný každou stranou zvlášť a jeho úkolem je identifikace dialogu. Po dobu dialogu se nemění a je přidáván k polím **From** a **To**.

Pole **Contact** obsahuje URI uživatele pro přímé spojení bez serverů. Toto pole je povinné u zpráv INVITE, SUBSCRIBE, NOTIFY.

Pole **Call-ID** jednoznačně identifikuje hovor. Je vytvořen náhodně vygenerovanými znaky, které doplňuje @ a vlastní doménová adresa (jméno). Pole musí být identické pro všechny zprávy v dialogu. Společně se **From** a **To** jednoznačně identifikují dialog.

Pole **CSeq** obsahuje sekvenční číslo a název metody. Číslo se s každou další žádostí inkrementuje, jedině v případě ACK a CANCEL tomu tak není.

Pole **Content-Type** identifikuje typ média obsaženého v těle. Musí být uvedeno, pokud tělo není prázdné. V tomto případě se jedná o hlavičku SDP paketu.

Pole **Content-Length** určuje velikost těla zprávy. Jestliže zpráva nemá tělo, je hodnota nastavena na 0. Uvedený příklad obsahuje v těle zprávy SDP hlavičku pro vyjednávání kodeků.

Podrobnější informace o SDP protokolu jsou uvedeny v kapitole 2.5.5.

Po doručení zprávy INVITE volanému uživateli, ten posílá zpět odpověď 100 Trying značící pokus o spojení. Poté přichází k volajícímu odpověď 180 Ringing indikující začátek vyzvánění u volaného uživatele. Jestli byla žádost kladně vyhodnocena a uživatel hovor přijal, posílá zprávu 200 OK:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.10.1.99:5060;branch= z9hG4bK343bf628
;received=100.101.102.103
From: "Test 15" <sip:15@10.10.1.99>;tag=as58f4201b
To: <sip:13@10.10.1.13>;tag=bf48a32hdk
Call-ID: 326371826c80e17e6cf6c29861eb2933@10.10.1.99
CSeq: 1 INVITE
Contact: <sip:13@10.10.1.13>
Content-Type: application/sdp
Content-Length: 145

v=0
o=root 1821 1821 IN IP4 10.10.1.99
s=session
c=IN IP4 10.10.1.99
t=0 0
m=audio 10567 RTP/AVP 0 8 101
a=rtpmap:0 PCMU/8000
```

Zpráva 200 OK je v podstatě kopie zprávy INVITE pouze se změnou v některých polích i parametrech. První řádek nám značí, že jde o kladnou odpověď. Je třeba si všimnout, že se nezaměnily hodnoty polí **From** a **To**. Zpráva 200 OK je odpověď na žádost, proto k záměně nedochází. V těle zprávy (SDP hlavička) se nám zobrazily možnosti odpovídajícího uživatele. Aby došlo k úspěšnému spojení, je odeslána potvrzující zpráva ACK:

```
ACK sip:13@10.10.1.13 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.99:5060;branch=z9hG4jP9A53428A2St8
```

From: "Test 15" <sip:15@10.10.1.99>;tag=as58f4201b
To: <sip:13@10.10.1.13>;tag=bf48a32hdk
Call-ID: 326371826c80e17e6cf6c29861eb2933@10.10.1.99
CSeq: 1 ACK
Content-Length: 0

V této zprávě se změna projevila u prvního řádku a pole **CSeq**, kde byla změněna metoda. Navíc byla změněna hodnota parametru **branch**, byl vygenerován nový kód, jelikož zpráva ACK se považuje za novou transakci. Zpráva ACK potvrzuje zprávu INVITE. Tímto krokem začíná relace a uživatelé si vyměňují data. Jestliže se jeden z uživatelů rozhodne spojení ukončit, odešle zprávu BYE druhému uživateli. Zpráva může mít následující tvar:

BYE sip:15@10.10.1.99 SIP/2.0
Via: SIP/2.0/UDP 10.10.1.13:5060;branch=z9hG4bKe05d543r5w
Max-Forwards: 70
To: "Test 15" <sip:15@10.10.1.99>;tag=as5ru752u9
From: <sip:13@10.10.1.13>;tag= bf48a32hdk
Call-ID: 326371826c80e17e6cf6c29861eb2933@10.10.1.99
CSeq: 1 BYE
Contact: <sip:15@10.10.1.99>
Content-Length: 0

I když bylo pole **Via** změněno, podle pole **Call-ID** a parametru **tag** se určí, že zpráva BYE je součástí jednoho dialogu. Ke správnému ukončení spojení je nutné odeslat zprávu 200 OK, která potvrzuje zprávu BYE.

Uvedená komunikace je jenom teoretická, v praxi se setkáváme s prvky jako Proxy servery, Registrační servery, které komunikaci ztěžují. Mohou se objevit i SIP brány, které připojují SIP uživatelé do klasické telefonní sítě.

2.5.3 Jiné typy zpráv

Protokol SIP není omezen jen těmito třemi typy zpráv, ale k dispozici máme další metody, které dokážou upravovat parametry hovoru.

V tabulce 2.1 jsou uvedeny často používané metody a jejich stručné vysvětlení. Podrobnější informace k metodám a jejich funkci lze nalézt v [15].

Tab. 2.1: Další významné metody

Název zprávy	Stručný popis
REGISTER	Používá se pro registraci/přihlášení klienta k serveru.
CANCEL	Používá se pro ukončení spojení ještě před přijetím zprávy 200 OK. Jinak se spojení ukončuje zprávou BYE.
UPDATE	Slouží ke změně parametrů hovoru ještě než je přijata zpráva ACK. Jinak se použije nová zpráva INVITE.
MESSAGE	Slouží pro přenos tzv. rychlých zpráv (instant message).
OPTIONS	Zjišťuje možnosti serveru nebo jiného klienta.
NOTIFY	Slouží k upozornění na určitou událost (např. připojení jiného klienta).

2.5.4 Odpovědi

Ve zmíněné ukázce principu komunikace už byla použita jedna z mnoha odpovědí a to 200 OK. Protokol SIP dělí odpovědi do několika skupin podle toho, jakou funkci zastávají. Každá skupina se označuje vlastním prvním číslem. Zbylá dvě čísla blíže přibližují hodnotu odpovědi.

1xx - Informativní

Zpráva Informativní značí, že žádost byla přijata druhým agentem a ten ji zpracovává. Zprávy 1xx v tabulce 2.2 nejsou konečné, výsledek zpracování vyslaného požadavku není ještě znám.

Tab. 2.2: Odpovědi 1xx

Kód	Slovní popis	Stručný popis
100	Trying	Odesláno okamžitě zpět, jakmile uživatel přijal metodu INVITE.
180	Ringing	U klienta započalo vyzvánění.
181	Call Is Being Forwarded	Upozorňuje na přesměrování hovoru.
182	Queued	Požadavek byl vložen do fronty.
183	Session Progress	Informuje o stavu hovoru.

2xx - Úspěšné

Tyto odpovědi značí, že zpráva byla úspěšně zpracována. Jedna z těchto dvou odpovědí v tabulce 2.3 je zároveň poslední odpověď, kterou odesílatel dostane na svoji žádost.

Tab. 2.3: Odpovědi 2xx

Kód	Slovní popis	Stručný popis
200	OK	Potvrzuje úspěšnou žádost.
202	Accepted	Oznamuje, že žádost byla přijata, ale nebyla splněna.

3xx - Přesměrování

Tyto odpovědi jsou logicky často posílány Serverem přesměrování. I klienti, popřípadě servery, mohou posílat tyto zprávy, jen musí mít přednastavené přesměrování hovorů.

Tab. 2.4: Odpovědi 3xx

Kód	Slovní popis	Stručný popis
300	Multiple Choice	Uživatel se bude pokoušet spojit s adresami uvedených v polích Contact , kterých může být ve zprávě několik.
301	Moved Permanently	Uživatel se nalézá trvale na jiné adrese. Tato adresa může být uložena pro příští akce.
302	Moved Temporarily	Uživatel se nalézá dočasně na jiné adrese. Tato adresa se nemá ukládat pro příští akce.
305	Use Proxy	Pro zjištění adres se kontaktuje Proxy server.

4xx- Chyba u klienta

Žádost nebylo možné zpracovat, server ji nepochopil. Zpráva měla pravděpodobně nesprávný formát. Příčina chyby je uvedena v odpovědi. V této kategorii je okolo třiceti možností, proto je nesmyslné vypisovat všechny. Uvedené odpovědi v tabulce 2.5 jsou jedny z nejběžnějších. Více informací pro tuto kategorii lze nalézt v [13].

Tab. 2.5: Odpovědi 4xx

Kód	Slovní popis	Stručný popis
400	Bad Request	Zpráva neměla správný formát (některé povinné pole nebylo obsaženo ve zprávě).
401	Unauthorized	Žádost potřebuje ověření uživatele.
403	Forbidden	Žádost byla pochopena, ale nebyla z nějakého důvodu splněna.
404	Not Found	Server zjišťuje, že se uživatel nenachází na uvedené adrese.
407	Proxy Authentication Required	Podobná kódu 401, ale indikuje potřebu autentizace se Proxy serverem. Využívá se například u telefonní brány.
408	Request Timeout	Vypršela doba pro zpracování žádosti.
486	Busy Here	Klient byl kontaktován, ale odesílá odpověď obsazeno.
491	Request Pending	Zabraňuje, aby oba uživatelé změnili současně parametry hovoru.

5xx - Chyba na serveru

Odpovědi v této kategorii jsou způsobeny chybou na straně serveru. Odpověď většinou obsahuje pole **Retry-After** s časovým intervalem, který po uplynutí povoluje opakovat odeslání žádosti.

Tab. 2.6: Odpovědi 5xx

Kód	Slovní popis	Stručný popis
500	Server Internal Error	Informuje o vnitřní chybě serveru.
501	Not Implemented	Zpráva obsahuje neznámou metodu.
502	Bad Gateway	Server sloužící jako brána do jiné sítě oznamující chybu v druhé síti.
503	Service Unavailable	Server je dočasně nedostupný.
504	Gateway Timeout	Server sloužící jako brána oznamuje vypršení doby v jiné síti.
505	Version Not Supported	Server nepodporuje verzi SIP protokolu užitého v žádosti.
513	Message Too Large	Zpráva byla příliš dlouhá pro zpracování.

6xx - Globální chyba

Server používá tyto odpovědi, když zpráva nemůže být zpracována nikde v celé síti. V tomto případě by uživatel neměl opakovat žádost, dokud neuplyne doba v poli `Retry-After`.

Tab. 2.7: Odpovědi 6xx

Kód	Slovní popis	Stručný popis
600	Busy Everywhere	Spojení nebylo uskutečněno, uživatel odesílá obsazeno.
603	Decline	Odpověď neobsahuje důvody zamítnutí žádosti.
604	Does Not Exist Anywhere	Server odesílá odpověď v případě, že adresa není nikde v síti.
606	Not Acceptable	Uživatel byl kontaktován, ale parametry ke spojení nevyhovují oběma koncům relace.

2.5.5 Protokol SDP

Protokol SDP slouží k určování parametrů přenosu, popisu kódování a popisu relace a je popsán v RFC 4566 [16]. Tělo je tvořeno jednopísmenným příznakem a jeho vlastností ve tvaru `typ hodnota`.

Zde je částečný výběr parametrů:

- v - verze SDP protokolu
- o - tvůrce a identifikátor relace
- s - jméno relace
- c - cílová adresa spojení
- t - čas začátku a konce relace
- m - informace o mediu, tok dat, číslo portu a transportní protokol
- a - doplňkové informace
- i - informace o relaci
- r - čas opakování

2.6 WebSocket protokol

WebSocket API je standardizován W3C. WebSocket standardizoval IETF jako RFC 6455 [17]. WebSocket je navržen pro implementaci do webových prohlížečů, ale je nutné, aby tuto technologii podporovaly i servery. Uspodňuje zobrazení živého obsahu a hraní real-time her. Mají-li být poslána data mezi klientem a serverem v reálném čase, je nutné vytvořit mezi oběma konci stabilní spojení pomocí WebSocket protokolu.

V kombinaci s WebSocket API je umožněna obousměrná komunikace z webové stránky na vzdálený server pomocí jednoho připojení. Komunikace probíhá pomocí TCP na portech HTTP 80 nebo 443 (využití Transport Layer Security - TLS protokolu TCP), takže není omezen firewall. Toto se využívá u signálního protokolu SIP, který potřebuje mít stále spojení pro posílání zpráv.

Jakmile klient a server odeslali úspěšně své zahajovací zprávy (handshake), může začít přenos dat. Byl založen obousměrný komunikační kanál, kde každá strana může posílat data podle své vůle.

Existují datové rámce obsahující binární hodnoty, které si vlastní aplikace dekoduje a kontrolní rámce, které určují stav připojení WebSocket protokolu. V případě, že klient bude ukončovat WebSocket spojení, odešle jeden ze Status kódů:

Tab. 2.8: Rezervované rozsahy Status kódů

Kód	Definice
0 – 999	Rozsah se nepoužívá.
1000 – 2999	Určeno pro definici WebSocket protokolu a budoucí trvalá rozšíření.
3000 – 3999	Použití knihoven, rámců, aplikace. Jsou registrovány v IANA.
4000 – 4999	Rozsah je vyhrazen pro soukromé použití.

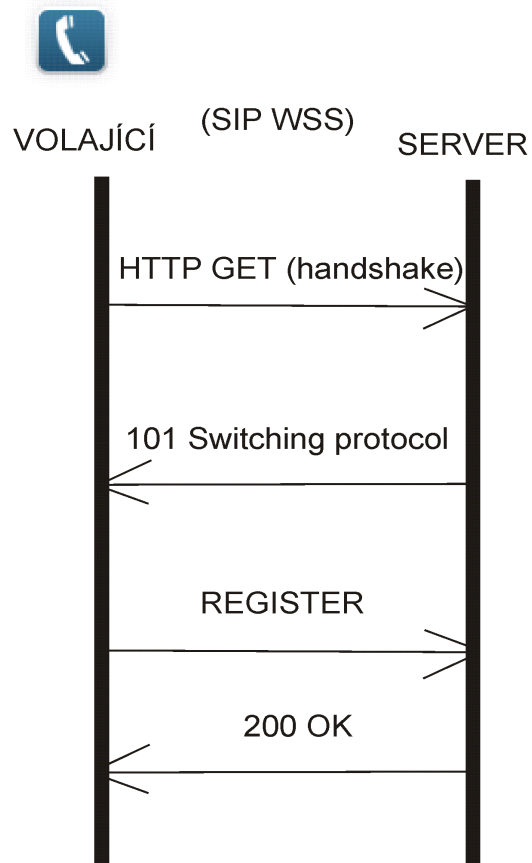
Tab. 2.9: Definované Status kódy WebSocket protokolu pro uzavření rámce

Kód	Definice
1000	Standardní uzavření, účel připojení byl splněn.
1001	Prohlížeč přešel na jinou stránku.
1002	Ukončení spojení. Vyskytla se chyba protokolu.
1003	Ukončení spojení. Typ příchozích dat nelze přijímat.
1004	Kód je vyhrazen pro budoucí využití.
1005	Nesmí být nastaven jako status kód pro uzavření rámce. (používá se v aplikacích očekávajících signalizaci, žádný status nebyl přítomen)
1006	Nesmí být nastaven jako status kód pro uzavření rámce. (používá se v aplikacích očekávajících signalizaci, spojení nebylo uzavřeno obvyklým způsobem)
1007	Ukončení spojení. Byla přijata data, která nebyla v souladu s typem zprávy.
1008	Ukončení spojení. Byla obdržena zpráva, která porušuje WebSocket politiku.
1009	Ukončení spojení. Byla obdržena zpráva, která je příliš velká.
1010	Ukončení spojení. Server vyjednával rozšíření, ale v zahajovací zprávě se změny neuvedly.
1011	Ukončení spojení. Nebyla splněna podmínka na splnění požadavku.
1015	Nesmí být nastaven jako status kód pro uzavření rámce. (používá se v aplikacích, při neplnění zahajovacích zpráv TLS nebo certifikát serveru nelze ověřit)

Definice Status kódů byla zjednodušena pro lepší pochopení. Bližší informace lze nalézt v [17].

2.6.1 WebSocket SIP podprotokol

Pro použití WebSocket SIP „podprotokolu“ musí klient uvést v zahajovací zprávě v poli `Sec-WebSocket-Protocol` hodnotu `sip`. Je nutné, aby server zprávu opětoval a hodnota ve `Sec-WebSocket-Protocol` korespondovala.



Obr. 2.5: Registrace uživatelů přes WebSocket protokol.

Zahajovací zpráva od klienta vypadá následovně:

```

GET / HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ ==
Origin: http://example.com
Sec-WebSocket-Protocol: sip
Sec-WebSocket-Version: 13
  
```

Odpověď ze serveru vypadá následovně:

```

HTTP/1.1 101 Switching Protocol
Upgrade: websocket
Connection : Upgrade
Sec-WebSocket-Accept: s3pLMBiTxaQ9kYGzzhZRbK + x0o =
Sec-WebSocket-Protocol: sip
  
```

WebSocket zprávy se mohou přenášet buď v textových, nebo v binárních rámcích. SIP protokol umožňuje obě verze ve svých zprávách. Klient a server vybírají variantu, kterou použijí. Každá SIP zpráva je obsažena v rámci jednoduché WebSocket zprávy a v této zprávě může být umístěna pouze jediná. Po úspěšné výměně zahajovacích zpráv se můžou začít přenášet SIP zprávy. Část kódu SIP protokolu (registrace k SIP serveru), jež je implementována v protokolu WebSocket:

```
REGISTER sip:13@10.10.1.13 SIP/2.0
```

```
Via: SIP/2.0/WSS 10.10.1.99:5060;branch=z9hG4bK343bf628
```

```
SIP/2.0 200 OK
```

```
Via: SIP/2.0/WSS 10.10.1.99:5060;branch= z9hG4bK343bf628
```

Obě zprávy mají v transportním protokolu uvedeno WSS. Toto značí přenos přes SIP WebSocket zabezpečený sub-protokol a určuje rozdíl oproti běžnému SIP protokolu. Klient se serverem (popřípadě s klientem) si vyměňují zprávy INVITE, TRYING, RINGING, 200 OK, ACK,...

3 NÁVRH VLASTNÍHO ŘEŠENÍ

3.1 Cíl práce

Hlavním cílem této práce je vytvořit plnohodnotného SIP klienta umožňujícího uskutečnit nejen videohovor propojující obě strany, ale i zasílat mezi oběma stranami rychlé zprávy tzv. instant message. Navíc po připojení k serveru, který nám toto spojení zrealizuje, se zobrazí seznam kontaktů zaregistrovaných k serveru.

Výsledná aplikace je realizována jazyky JavaScript a PHP. Dále je zde použita technologie Ajax a kaskádové styly. To by mělo zaručit správnou funkčnost mezi všemi platformami. Avšak musí být splněno několik požadavků. Prvním z nich je povolený jazyk JavaScript ve webovém prohlížeči. Dále na serveru musí být nainstalována podpora jazyku PHP.

Aplikace byla vyvíjena výhradně pro webový prohlížeč Chrome od společnosti Google z důvodu plné podpory technologie WebRTC. Tuto technologii podporuje i vývojová verze prohlížeče Firefox Nightly od společnosti Mozilla.

Jako SIP server byl použit volně stažitelný program OfficeSIP ve verzi 3.4, který pracuje pod operačním systémem Windows a zajišťuje možnost registrace klienta. Dále umožňuje výměnu SIP zpráv a multimediálních dat při videohovoru. Tento projekt je celosvětově používán pouze v malém množství. Některé technologie jsou ještě ve vývoji a nejsou oficiálně schváleny. Proto také podpora těchto technologií je nedostatečná pro důsledné testování.

Následující body shrnují požadavky kladené na výslednou aplikaci.

- Možnost registrace nového uživatele k serveru
- Možnost přihlášení/odhlášení uživatele k serveru
- Načtení uživatelů, kteří se k serveru registrovali, tudíž mohou být osloveni
- Odeslání a přijetí videohovoru/textové zprávy
- Bezchybnost a stabilita aplikace

3.2 Knihovny

Sestavení vlastního SIP klienta je závislé na použité knihovně. Je možnost vytvoření vlastní knihovny nebo můžeme využít některou z dostupných knihoven, které jsou k dispozici na Internetu.

Výběr správné SIP knihovny se stal jedním z nejdůležitějších kroků návrhu SIP klienta. Zásadním prvkem pro výběr použité knihovny se stal zdrojový kód.

3.2.1 Flashphoner

Jedná se o volně přístupnou knihovnu ¹ vývojového týmu sídlícího v Ruské státní Novosibiřské univerzitě. K dispozici byly poskytnuty knihovny JavaScript a Flash. Serverová aplikace je napsána v jazyce Java, místy i v jazyce C. Aplikace pro klienta je napsaná v jazyce JavaScript a v jazyce ActionScript.

Tato aplikace je ale poměrně drahá. K dispozici existují tři licence, které jsou odstupňovány podle určených využití. V základní verzi jsou omezeny značně, naopak v neomezené verzi jsou k dispozici všechny dostupné funkce aplikace.

Pro své využití podporuje velkou část serverů. Například celosvětově využívaný server Asterisk nebo Wowza Media Server 2, pro který byla aplikace vyvíjena.

Součástí projektu je i rozsáhlá přehledná dokumentace. Knihovna Flash je pro tento projekt bezvýznamná, jelikož technologie Flash se nebude používat. Knihovna Javascriptu byla převzata od Doubango Telecom. Jelikož bylo složité se v knihovně vyznat, rozhodl jsem se tuto knihovnu nevyužít.

3.2.2 sipML 5

Projekt sipML5 ² je volně dostupný a byl vytvořen společností Doubango Telecom se sídlem v Paříži. Zdrojové kódy jsou napsány v jazyce JavaScript a celý zdrojový kód je příkladně uspořádán, proto se orientace v něm jeví jednoduše. Knihovna je dělena na velké množství skriptů, které jsou pojmenovány podle své funkce. Pro pochopení je počet a obsah komentářů popisujících funkce kódu nedostatečný.

Na webových stránkách je k dispozici fórum pro otevřené diskuze o projektu a knihovně. Knihovna má licenci GPL v3, což znamená svobodu čtení a změny obsahu. Podpora bezplatných serverů Asterisk a Kamailio je zajištěna.

Na webových stránkách projektu je zveřejněna textová verze knihovny v systému SVN, která je volně upravitelná pro nahlášení chyb vývojářům. Je možnost stáhnutí celé knihovny přes doporučený program Tortoise SVN, což je nástroj pro práci s verzovacím systémem SVN.

Zejména kvůli nedostatečnému počtu komentářů a popisu funkcí jsem se rozhodl knihovnu nepoužít.

3.2.3 JsSIP

Za touto veřejně dostupnou Javascript knihovnou stojí J. L. Millán, I. B. Castillo a S. I. Corretgé. Celá knihovna je napsaná v jazyce JavaScript. Obsahuje třídy pro plnohodnotné SIP spojení. Popisky funkcí jsou kvalitně okomentovány. Na webových

¹Knihovna je umístěna na webových stránkách: <http://flashphoner.com/>

²Více informací lze nalézt na webových stránkách: <http://sipml5.org/index.html?svn=13>

stránkách projektu ³ je rozsáhlá dokumentace ke všem třídám a rozepsané možné chyby a výjimky.

U knihovny je použita licence MIT, která umožňuje jakékoliv nakládání s obsahem, pouze text licence musí být dodán spolu s obsahem. Knihovna spolupracuje s většinou používaných serverů, například Kamilio, Asterisk, OfficeSIP nebo vlastní OverSIP.

Pro tuto práci je vhodným kandidátem knihovna JsSIP, která má široké využití a implementace do aplikace bude jednodušší než u ostatních knihoven. Obsahuje třídu pro WebSocket spojení a jsou zahrnuty SIP zprávy. Sami autoři podtrhují její malou velikost. Jedná se asi o třikrát menší knihovnu vzhledem ke zbývajícím dvěma.

3.2.4 Výběr dalších knihoven

V projektu jsou ještě použity knihovny jQuery a jQuery-FancyBox.

Knihovna jQuery je framework pro tvoření webových stránek. Ulehčuje a rozšiřuje možnosti práce v jazyku JavaScript jednoduššími postupy.

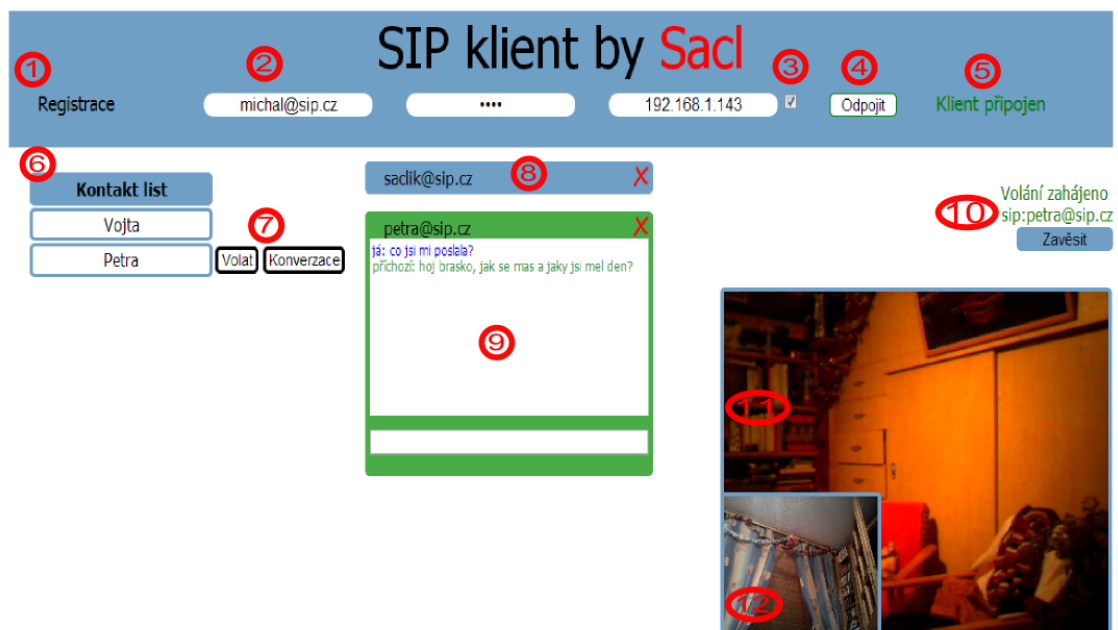
Knihovna jQuery-FancyBox je nástroj pro vytvoření vyskakovacích oken (v projektu je použita při příchozím hovoru). Byla vytvořena především pro zobrazování obrázků nad hlavním oknem. Možnosti ale dovolují vytvořit HTML obsah, který bude zobrazován. Vyskakovací okno může být opatřeno i specifickými vlastnostmi, které určují chování celého okna.

3.3 Grafické prostředí

Ke stránce programové existuje i stránka grafického rozhraní. Celé grafické prostředí bude vytvořeno s využitím jazyka HTML, JavaScript a PHP. Z úvodní stránky je možnost se přepnout na webovou stránku určenou pro registraci k serveru. Po zadání všech povinných polí a ověření, není-li už SIP adresa je obsazena, registrace proběhne a uživatel je přesměrován zpět na úvodní stránku.

Zde se vyplní tři pole uživatelskými údaji a proběhne přihlášení. Po přihlášení na webové stránce přibude seznam kontaktů, který obsahuje přezdívky všech zaregistrovaných uživatelů. V tomto okamžiku je webová stránka připravena pro přímou zahájení videohovorů nebo výměnu rychlých zpráv. Celkové grafické zpracování je zobrazeno na obr. 3.1, jehož jednotlivé body jsou popsány pod obrázkem.

³Webové stránky projektu JsSIP: <http://www.jssip.net/documentation/0.2.x/api>



Obr. 3.1: Grafické prostředí SIP klienta

1. Tlačítko pro registraci klienta
2. Pole pro zápis SIP adresy, Hesla, a WS serveru
3. Tlačítko pro zvolení výchozí IP adresy WebSocket serveru
4. Tlačítko Připojit/Odpojit pro přihlášení/odhlášení klienta od serveru
5. Stavové okno ukazující stav připojení
6. Seznam kontaktů s přezdívkami zaregistrovaných klientů
7. Tlačítka Volat a Konverzace pro uskutečnění videohovoru, popř. zobrazení chatovacího okna
8. Chatovací okno v minimalizovaném stavu
9. Chatovací okno v otevřeném stavu při přijaté nové zprávě
10. Informační lišta se jménem druhého účastníka a tlačítko pro zavěšení hovoru
11. Video okno zobrazující vzdáleného klienta
12. Video okno zobrazující lokálního klienta

3.4 Architektura

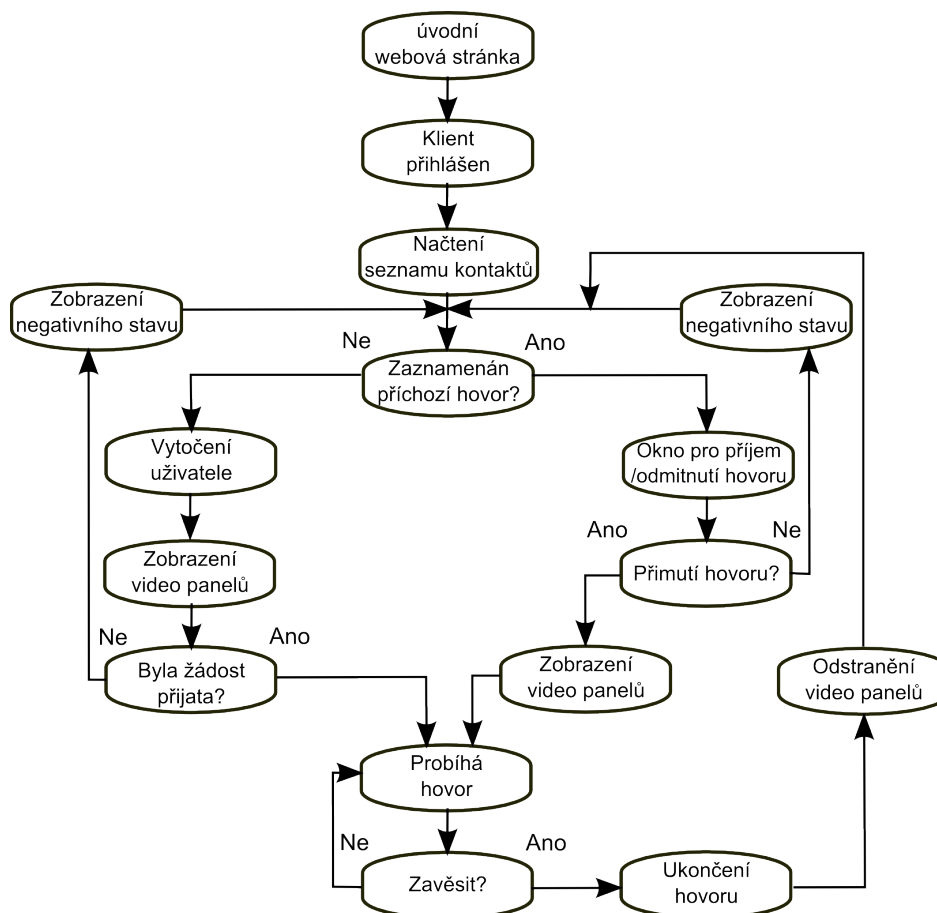
Při realizaci aplikace byly vytvořeny vývojové diagramy vyznačující postupné řízení programu..

Na obrázku 3.2 je vyobrazena funkce videohovorů. Diagram je zahájen samotným přihlášením k SIP serveru a načtením dat ze serveru.

Při vytočení vzdáleného uživatele jsou povolována práva pro přístup k webkameře a mikrofonu. Je možnost tyto práva v prohlížeči trvale aktivovat. Při odmítnutí se hovor neuskuteční a celá relace je ukončena. Při povolení je odeslána žádost o hovor vzdálenému uživateli a je očekávána odpověď.

Při příchozím hovoru je zobrazeno aktivní okno na zašedlém pozadí s jménem volajícího uživatele a možnostmi přijmout nebo odmítnout hovor. Při přijmutí hovoru se opět povolují práva na pořízení obsahu. Při kladné odpovědi už probíhá samostatný přenos multimediálních dat.

Videohovor je možné ukončit z obou stran relace. Poté je zobrazen ve webové stránce na obou stranách status o ukončeném volání a videopanely jsou skryty.



Obr. 3.2: Architektura videohovoru

Na obrázku 3.3 je zobrazeno chování webové stránky při tvorbě a příjmu rychlých zpráv.

Obě možnosti, videohovory i rychlé zprávy pracují na webové stránce společně a vzájemně se neovlivňují jejich funkce.

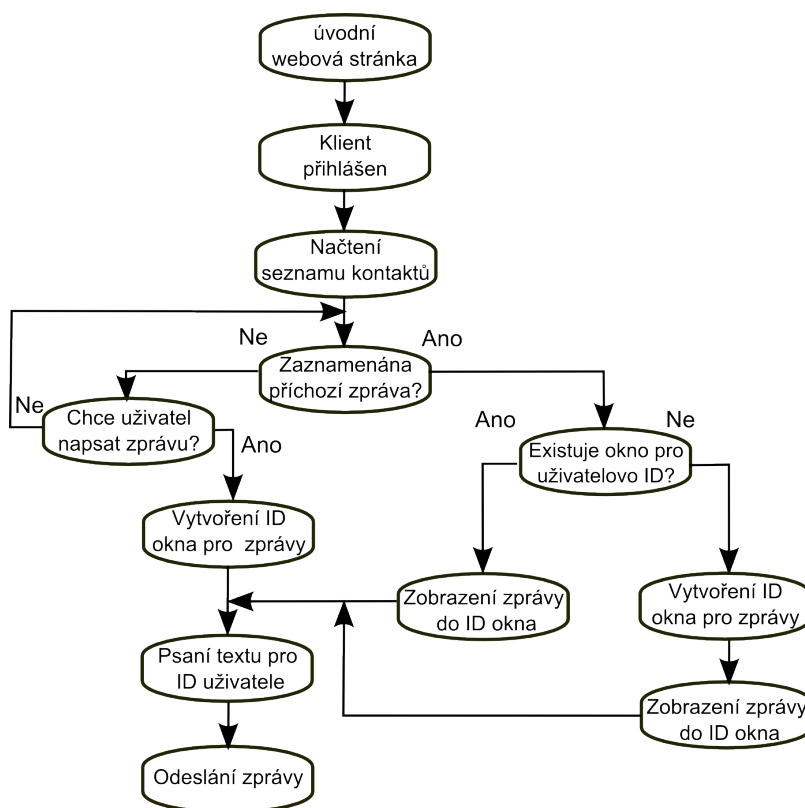
V případě posílání rychlých zpráv jsou k tomu určena konverzační okna, která jsou vytvořena po stisku tlačítka Konverzace. Toto okno je identifikované SIP adresou, která jasně určuje s kým je tato konverzace vedena. Těchto konverzačních oken může být na webové stránce neomezeně.

Ovšem pro každého vzdáleného uživatele je k dispozici pouze jediné, a proto není možné, aby zpráva byla odeslána jinam, než kam bylo plánováno.

Pro lepší orientaci v konverzačních oknech je možnost jednotlivá okna minimalizovat (data se zachovají) nebo zavřít (konverzační okno a vnořená data se vymažou).

Po přijetí zprávy od vzdáleného uživatele je porovnáváno, jestli už webová stránka neobsahuje otevřené konverzační okno pro konkrétního uživatele. Pokud webová stránka neobsahuje toto konverzační okno, je nově vytvořeno a příchozí zpráva se vypíše v něm.

Pokud už konverzační okno bylo vytvořeno dříve, je příchozí zpráva zobrazena v příslušném okně.



Obr. 3.3: Architektura rychlých zpráv

3.5 Popis jednotlivých funkcí

3.5.1 Registrace k serveru

Registrace probíhá v jazyce PHP 3.1, který si naplňuje čtyři proměnné daty ze stránky. Tyto proměnné připisuje v určitém tvaru `$userdata` do souboru umístěném na serveru. Po splnění úkolu je stránka přesměrována na úvodní stránku `index.php`.

```
<?php
$firstName = $_REQUEST["Login"];
$password = $_REQUEST["Password"];
$nickName = $_REQUEST["nickName"];
$email = $_REQUEST["Email"];

$userdata = "\r\n$firstName,$password,$nickName,$email,,,,,,,,,";
$soubor = fopen("C:/ProgramData/OfficeSIP/Server/Accounts/00
00001/Users.csv", "a+");

fwrite($soubor, $userdata);
fclose($soubor);
if ($userdata) { header("HTTP/1.1 301 Moved Permanently");
    header("Location: index.php");
    header("Connection: close");
}
?>
```

Výpis 3.1: Zdrojový kód registrace k SIP serveru

Na stejném principu je vytvořeno ověřování dostupných volných SIP adres, kde pomocí technologie AJAX je odeslána hodnota v poli `Login`. V otevřeném souboru uživatelů na serveru se hledá shoda v poli `Login` a vybraným sloupcem tabulky.

3.5.2 Přihlášení k serveru

Před přihlášením je k dispozici zaškrtačací tlačítko, kde pomocí PHP skriptu

```
<?php echo $_SERVER['SERVER_NAME'] ?>
```

je vypsáno do pole IP WS server automaticky jméno serveru, v tomto případě IP adresu. Funkce `phoneInit()` vytváří připojení k serveru.

Pomocí této funkce jsou odeslány přihlašovací údaje k serveru a je očekávána odpověď. Pokud je vrácena hodnota `registered`, funkce `createContactList()` je odkázána na vytvoření seznamu kontaktů pomocí technologie AJAX. Dále je

na webové stránce zobrazen text značící úspěšné připojení viz. následující výpis 3.2. K odhlášení ze serveru je určena funkce `phoneStop()`.

```
phone.on('registered', function(e){
    createContactList();
    console.info('Registered');
    writeStatus("Klient připojen");
    $('#contact_list').attr('hidden',false);
});
```

Výpis 3.2: Vykonávaná funkce po přihlášení

Prvek seznam kontaktů je vytvářen PHP skriptem, tudíž na straně serveru 3.3. Nejprve je vytvořena hlavička tabulky. Poté je otevřen serverový soubor se zaregistrovanými uživateli a data jsou vypsána do řádků tabulky. Pomocí AJAXu je získána SIP adresa přihlášeného uživatele, která je porovnávána s hodnotami v řádku, aby nebyla zobrazena v seznamu kontaktů vlastní přezdívka. Zobrazovaný řádek obsahuje pouze jedno pole a tím je Přezdívka a dvě skrytá tlačítka Volat a Konverzace. Poté je soubor uzavřen.

```
<?php
//Záhlaví seznamu kontaktů
...
$vlastSIP = $_GET["data1"];
//Otevření souboru a vyčtení dat
$soubor = fopen("C:/ProgramData/OfficeSIP/Server/Accounts/00
000001/Users.csv", "r");
$data = File("C:/ProgramData/OfficeSIP/Server/Accounts/00
000001/Users.csv");
//první řádek je prázdný
$druhy[0] = fgets($soubor);
//cyklus pro vyčítání dat z proměnné
for ($i = 1; $i < Count($data); $i++) {
    // další řádky
    $druhy[$i] = fgets($soubor);
    //rozložení celého řádku do více polí
    $pole = explode(",", $druhy[$i]);
    //Podmínka pro porovnání přihlášeného uživatele s databází
    if ($vlastSIP == "{$pole[0]}@sip.cz")
        ;
    else
        echo("
        //HTML znaky-vypsání řádku Přezdívka
```

```

        a tlačítka Volat a Konverzace
    ");}
fclose($soubor);
echo("</table>");
?>

```

Výpis 3.3: Tvorba prvku seznam kontaktů

3.5.3 Funkce k obsluze videohovorů

Pro odeslání požadavku k videohovoru je využita funkce `placeCall()`, jenž v sobě nese SIP adresu volaného uživatele. K přenosu potřebných SIP zpráv pro vytvoření multimediálního spojení je určena funkce `newRTCSession()` a obsahuje události spojené s hovorem (selhal, ukončen, začal 3.4, čekání).

Funkce `started` vypisuje oznámení o zahájeném spojení se SIP adresou vzdáleného uživatele a tlačítkem pro ukončení hovoru. Obsažena je i proměnná `rtcSession`, jež je definována v knihovně JsSIP.

```

call.on('started',function(e){
    writeCall("Volání zahájeno <br/>" + uri +
        "<br/><input type='text' class='Callbutton'
        onclick='hangUp();' value='Zavěsit' readonly/>");
    var rtcSession = e.sender;
});

```

Výpis 3.4: Vykonávaná funkce po zahájení videohovoru

Příchozí hovor je označen `'incoming'`. Toto spouští funkci `incomingCall(uri)`, která v sobě nese SIP adresu volajícího uživatele a je vyvolán prvek `jQuery-Fancybox`.

Pomocí funkce `hangUp()` je hovor ukončen a relace je zrušena. Pro ukončení hovoru ještě před navázáním spojení druhé strany je určena funkce `cancelCall()`.

Funkce `answerCall()`, v jejímž těle je podstatné volání funkce `call.answer(options);`, vytváří kladnou odpověď, tudíž je hovor přijat. Naopak funkcí `declineCall()` je hovor odmítnut.

3.5.4 Zprostředkování rychlých zpráv

Pro odesílání rychlých zpráv musí být nejdříve vytvořeno konverzační pole. Stiskem tlačítka Konverzace je provedena funkce `Chat(uri)` 3.5, kde se porovnává vytvoření nového konverzačního okna s existujícími. Obsahem funkce jsou i HTML prvky pro vytvoření konverzačního pole.

```
function Chat (uri){
    var chat = '\\HTML prvky vytvářející konverzační pole';
    if(document.getElementById(uri1)!=null){
        ;
    }else{
        if(uri!==$(".chaturi").text()){
            $(".chatContainer").append(chat);};
        };
    };
};
```

Výpis 3.5: Funkce pro vytvoření konverzačního okna

Ve vytvořeném konverzačním poli je použit prvek `form` 3.6, který je zprostředkovatelem rozhraní, kde je obsažen textový blok pro ukládání vlastních a příchozích zpráv (barevně rozlišeny). Dále je zde připraven řádek pro vlastní text určen k odeslání.

Hodnoty v proměnných `text` a `uri` určují vlastní text a SIP adresu vzdáleného živatele. Odeslání zprávy je prováděno funkcí `phone.sendMessage(uri, text)`; po stisku tlačítka Enter na klávesnici (prázdný řádek nelze odeslat). Dále je vlastní text přesunut do dialogového okna a řádek pro vlastní text je vyprázdněn.

V poslední části funkce je měřena výška obsahu dialogového okna pro případné přetáčení okna.

```
$(".form.addMessage").live("submit", function() {
    var text = $(this).parent().find(".message").val();
    var uri = $(this).parent().find(".chaturi").text();
    try {
        if (text !== "") {
            //odeslání rychlé zprávy
            phone.sendMessage(uri, text);
            //přidání zprávy do určeného okna
            $(this).parent().find(".conversation").append("<li class=\"message\">já: " + text + "</li>");
            $(this).parent().find(".message").val("");
        } catch(e){
            console.log(e);
            return;
        }
        var scroll = $(this).parent().find('.conversation').height();
        //scrollování okna při jeho přeplnění
        $(this).parent().find(".scrollBox").parent().scrollTop(scroll);
    };
});
```

Výpis 3.6: Prvek form a vložené funkce

Za přenos rychlých zpráv je odpovědná funkce `newMessage()`, kde obsahem je SIP adresa posílajícího uživatele, SIP adresa druhé strany, data tedy vlastní text. V okamžiku přijetí zprávy je vyvolána funkce `incomingMessage(uri,text);`. Tato funkce je podobná funkci `Chat(uri)`, kde pokud ještě neexistuje konverzační okno, je vytvořeno jedno pro danou relaci a je naplněno přijatým textem.

Projekt obsahuje i další drobnosti v podobě minimalizace konverzačního okna, zavření okna a změny zabarvení okna při příchozí zprávě.

4 TESTY APLIKACE

Testování výsledné aplikace patří neodmyslitelně ke každému vývoji. Při samotném programování byla snaha o co největší eliminaci neošetřených stavů aplikace. V rámci práce je provedeno několik testů zaměřených především na přihlášení k serveru a odesílání SIP zpráv, videohovorů a rychlých zpráv na jejichž základě je předpokládán určitý scénář.

Testování bylo provedeno ve webovém prohlížeči Google Chrome verze 27, přesněji 27.0.1453.93. Dále byl použit webový prohlížeč Mozilla Firefox Nightly verze 23.0a1, tudíž vývojová verze standardního prohlížeče Mozilla Firefox.

Serverový operační systém byl instalován na virtuální stroj WorkStation verze 9.0.0 build-812388 od společnosti VMware.

SIP server OfficeSIP verze 3.4.rc byl instalován na operační systém Microsoft Windows Server 2008 R2 Standard x64bit.

4.1 Přihlášení klienta k SIP serveru přes prohlížeč Chrome

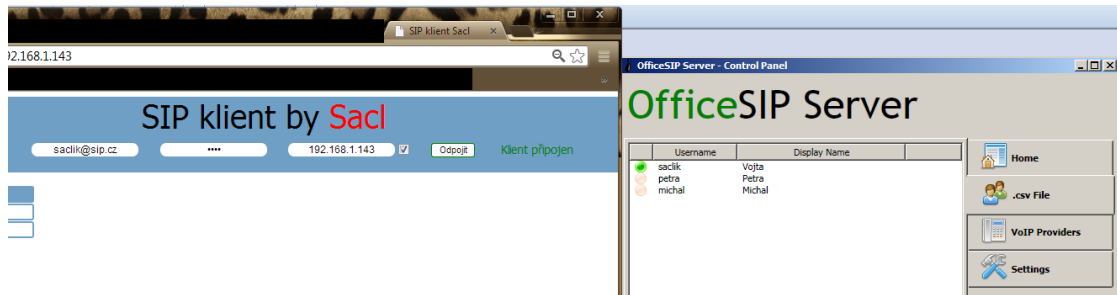
V novém okně webového prohlížeče byla načtena úvodní stránka projektu.

Po zadání potřebných přihlašovacích údajů byla data odeslána k serveru na zpracování. Na obrázku 4.1 je zobrazena webová stránka po přihlášení. Indikace přihlášení je značena zeleným nápisem **Klient Připojen**. Na straně serveru je indikace přihlášeného uživatele značena zeleným puntíkem v konzoli a indikace nepřihlášeného uživatele je barvou oranžovou.

Výpis 4.1 obsahuje celou komunikaci se SIP serverem během přihlášení. Konkrétně jsou to Registrační údaje, SIP zprávy REGISTER směrem k serveru, 401 Unauthorized ze směru od serveru, nová SIP zpráva REGISTER s autorizační složkou směrem k serveru a odpověď 200 OK pro potvrzení přihlášení.

Výsledek:

Test byl vykonán úspěšně bez jakékoliv známky chybovosti.



Obr. 4.1: Vyobrazení webové stránky v Chrome a konzole SIP serveru

JsSIP | UA | configuration parameters after validation:

- password: "...."
- register: true1
- registrar_server: sip:sip.cz
- ...
- uri: sip:saclik@sip.cz
- ws_servers: [{"ws_uri": "ws://192.168.0.103:5060", "sip_uri": "<sip:192.168.0.103:5060;transport=ws;lr>", ...
- hostport_params: "sip.cz"
- authorization_user: "saclik"

JsSIP | EVENT EMITTER | adding event newMessage, newRTCSession, registrationFailed, unregistered, registered, disconnected, connected

JsSIP | EVENT EMITTER | new listener added to event connected,

disconnected, newRTCSession, newMessage, registered,

unregistered, registrationFailed

JsSIP | UA | user requested startup...

JsSIP | TRANSPORT | connecting to WebSocket ws://192.168.0.103:5060

JsSIP | TRANSPORT | WebSocket ws://192.168.0.103:5060 connected

JsSIP | UA | connection state set to 0

JsSIP | EVENT EMITTER | emitting event connected

JsSIP | TRANSPORT | sending WebSocket message:

REGISTER sip:sip.cz SIP/2.0

Via: SIP/2.0/TCP 192.0.2.67;branch=z9hG4bK8643576

Max-Forwards: 69

To: <sip:saclik@sip.cz>

From: <sip:saclik@sip.cz>;tag=7jk7uvpgvd

Call-ID: enpt0fcioqdnhjpcn8q0am

CSeq: 81 REGISTER

Contact: <sip:26lcimv9@192.0.2.67;transport=ws>;reg-i...

Allow: ACK,CANCEL,BYE,OPTIONS,INVITE,MESSAGE

Supported: path, outbound, gruu
User-Agent: JsSIP 0.3.0
Content-Length: 0

JsSIP | TRANSPORT | received WebSocket binary message:
SIP/2.0 401 Unauthorized
Via: SIP/2.0/TCP 192.0.2.67;branch=z9hG4bK8643576
To: <sip:saclik@sip.cz>;tag=3c010ecdf783430d8f4d282f3b7a87bd
From: <sip:saclik@sip.cz>;tag=7jk7uvpgvd
Call-ID: enpt0fcioqdnhjpcn8q0am
CSeq: 81 REGISTER
x-Error-Details: No auth header
WWW-Authenticate: Digest realm="sip.cz",...
Content-Length: 0

JsSIP | TRANSPORT | sending WebSocket message:
REGISTER sip:sip.cz SIP/2.0
Via: SIP/2.0/TCP 192.0.2.67;branch=z9hG4bK1726418
Max-Forwards: 69
To: <sip:saclik@sip.cz>
From: <sip:saclik@sip.cz>;tag=7jk7uvpgvd
Call-ID: enpt0fcioqdnhjpcn8q0am
CSeq: 82 REGISTER
Authorization: Digest algorithm=MD5, username="saclik", realm="sip.cz", uri="sip:sip.cz",...
Contact: <sip:26lcimv9@192.0.2.67;transport=ws>;...
Allow: ACK,CANCEL,BYE,OPTIONS,INVITE,MESSAGE
...

JsSIP | TRANSPORT | received WebSocket binary message:
SIP/2.0 200 OK
Via: SIP/2.0/TCP 192.0.2.67;branch=z9hG4bK1726418
To: <sip:saclik@sip.cz>;tag=45220bab52a3404b846653bebe24b8e7
From: <sip:saclik@sip.cz>;tag=7jk7uvpgvd
Call-ID: enpt0fcioqdnhjpcn8q0am
CSeq: 82 REGISTER
Contact: <sip:26lcimv9@192.0.2.67;transport=ws>;+sip.instance=...
Expires: 599
Content-Length: 0
JsSIP | EVENT EMITTER | emitting event registered

Výpis 4.1: JavaScript konzole v Chrome

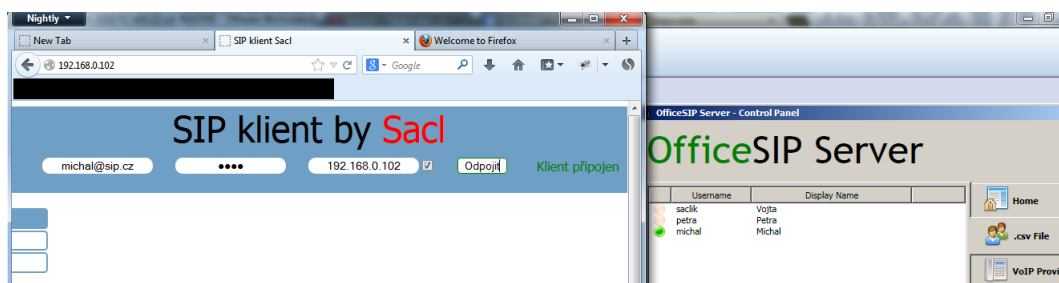
4.2 Přihlášení klienta k SIP serveru přes prohlížeč Firefox Nightly

Stejným způsobem jako v předešlém bodě je provedeno přihlášení. Na obrázku 4.2 je opět vyobrazena webová stránka a konzole SIP serveru. Webová stránka je v tomto případě otevřena ve webovém prohlížeči Firefox Nightly.

Obsah JavaScript konzole v prohlížeči Firefox Nightly je totožný s konzolí v prohlížeči Chrome.

Výsledek:

Test byl vykonán úspěšně bez jakékoliv známky chybovosti.



Obr. 4.2: Vyobrazení webové stránky Firefox Nightly a konzole SIP serveru

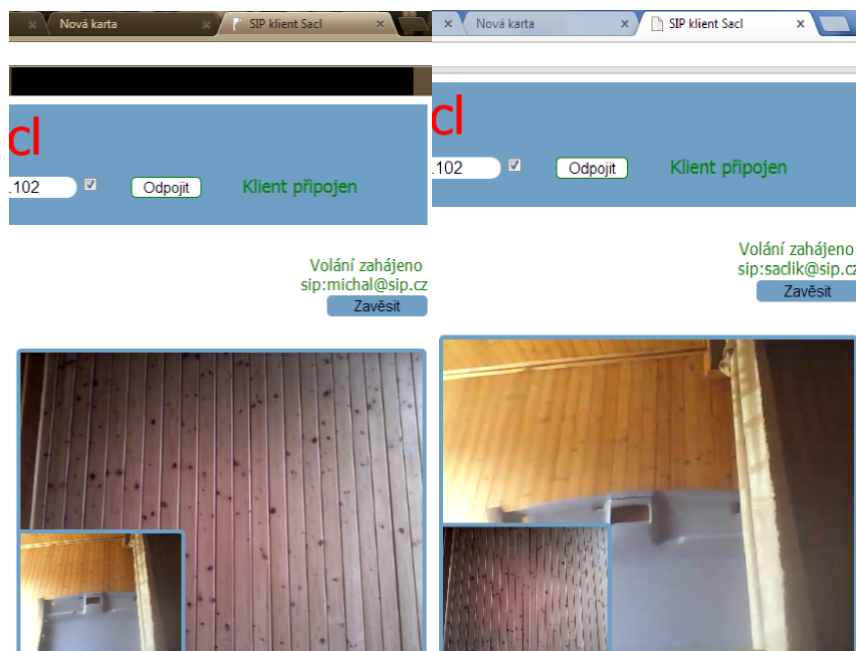
4.3 Videohovor uživatelů v prohlížeči Chrome

Oba vzdálení uživatelé byly úspěšně přihlášení. Poté byl jednou stranou vyslán požadavek na spojení. Druhá strana odpověděla kladně a videohovor byl započat. Výsledek je zobrazen na obrázku 4.3. Ve videohovoru byl obsažena i zvuková stopa.

Ve výpisu 4.2 je vypsána komunikace před a při zahájení videohovoru.

Výsledek:

Test byl vykonán úspěšně bez jakékoliv známky chybovosti.



Obr. 4.3: Videohovor dvou vzdálených uživatelů v prohlížeči Chrome

```

JsSIP | RTC SESSION | got local media stream
JsSIP | RTC SESSION | ICE candidate received: a=candidate:213...
JsSIP | RTC SESSION | ICE candidate received: a=candidate:355...
JsSIP | TRANSPORT | sending WebSocket message:
INVITE sip:saclik@sip.cz SIP/2.0
Via: SIP/2.0/TCP 192.0.2.239;branch=z9hG4bK1763179
Max-Forwards: 69
To: <sip:saclik@sip.cz>
From: <sip:michal@sip.cz>;tag=582u9cd4ir
...
v=0
o=- 2198865781 2 IN IP4 127.0.0.1
...
a=extmap:1 urn:ietf:params:rtp-hdext:ssrc-audio-le
JsSIP | TRANSPORT | received WebSocket binary message:
SIP/2.0 100 Trying
...
JsSIP | TRANSPORT | received WebSocket binary message:
SIP/2.0 180 Ringing
...
JsSIP | TRANSPORT | received WebSocket binary message:
SIP/2.0 200 OK

```

```
...
JsSIP | EVENT EMITTER | emitting event progress
JsSIP | TRANSACTION | Timer B expired for INVITE client ...
JsSIP | RTC SESSION | stream added: gfP...
JsSIP | DIALOG | dialog bp...lnv2m96kv changed to CONFIRMED state
```

Výpis 4.2: JavaScript konzole při navázání videohovoru v Chrome

4.4 Videohovor uživatelů v prohlížeči Ff Nightly

Oba vzdálení uživatelé byly úspěšně přihlášení. Poté byl jednou stranou vyslán požadavek na spojení. Po udělení oprávnění k přístupu k webkameře a mikrofonu byly zobrazeny ve webové konzoli dvě chyby. Tyto chyby, znázorněné ve výpisu 4.3, zapříčinili neodeslání požadavku vzdálenému klientovi.

Výsledek:

Test byl vykonán neúspěšně. Uživatelé nebyli spojeni a nebyl uskutečněn videohovor. Chyba nastala v knihovně JsSIP.

```
JsSIP | RTC SESSION | requesting access to local media
e.currentTarget is undefined
[Break On This Error]

...rtcMediaHandler.createAnswer(p,T)},d=function(){i.status!==a.STA
TUS_TERMINATED&&...

JsSIP | RTC SESSION | got local media stream
Value is not an object.
[Break On This Error]

...rtcMediaHandler.createAnswer(p,T)},d=function(){i.status!==a.STA
TUS_TERMINATED&&...
```

Výpis 4.3: JavaScript konzole při navázání videohovoru v Firefox Nightly

4.5 Videohovor uživatelů mezi prohlížeči Chrome a Firefox Nightly

Oba vzdálení uživatelé byly úspěšně přihlášení. Byly vyzkoušeny oba směry komunikace: tedy hovor z Google Chrome do Mozilla Firefox Nightly a poté z Mozilla

Firefox Nightly do Google Chrome. Opět byla ve webové konzoli v prohlížeči Firefox Nightly zobrazeny dvě chyby 4.3. Tyto chyby zapříčinili jak neodeslání požadavku vzdálenému klientovi, tak přijetí požadavku na spojení.

Výsledek:

Test byl vykonán neúspěšně. Uživatelé nebyli spojeni a nebyl uskutečněn videohovor. Chyba nastala v knihovně JsSIP.

4.6 Konverzace dvou uživatelů mezi prohlížeči Chrome a Firefox Nightly

V levé části obrázku 4.4 byla konverzace uskutečněna v prohlížeči Chrome. V pravé části byl SIP klient obsluhován v prohlížeči Firefox Nightly. Přijatá zpráva je označena zeleným písmem s popiskem **příchozí**:. Nepřečtená zpráva přebarví konverzační okno do zeleného odstínu pro snadnější orientaci při více otevřených konverzačních oknech.

Ve výpisu z JavaScript konzole 4.4 je vyobrazena komunikace SIP zpráv při přijmutí zprávy s obsahem: ?. Rychlé zprávy jsou posílány pomocí metody **MESSAGE**. Každá doručená zpráva je potvrzena zprávou 200 OK.

Výsledek: Test byl vykonán úspěšně bez jakékoliv známky chybovosti. Všechny zprávy byly doručeny správnému uživateli a byly zobrazeny.



Obr. 4.4: Konverzace dvou uživatelů v prohlížečích Firefox Nightly a Chrome

```
JsSIP | TRANSPORT | received WebSocket binary message:
MESSAGE sip:js8u66sp@192.0.2.53;transport=ws SIP/2.0
```

```

Via: SIP/2.0/WS i1728101125.invalid;branch=z9hG4bK0400000c
Max-Forwards: 68
To: <sip:saclik@sip.cz>
From: <sip:michal@sip.cz>;tag=8n0u5pdk86
Call-ID: ac2fgusig8b68vsavh7d
CSeq: 9010 MESSAGE
Content-Type: text/plain
Supported: path, outbound, gruu
User-Agent: JsSIP 0.3.0
Content-Length: 1

?
JsSIP | EVENT EMITTER | emitting event newMessage
JsSIP | TRANSPORT | sending WebSocket message:
SIP/2.0 200 OK
...

```

Výpis 4.4: JavaScript konzole při odeslání jedné zprávy

4.7 Konverzace tří uživatelů v prohlížečích Chrome

Tento scénář testuje zobrazení příchozí zprávy ve správném okně. Tedy, že zpráva zaslaná na určitou adresu je zobrazena v příslušném okně a v žádném jiném. Konverzační okna jsou oddělena a sama sebe neovlivňují, což je vidět i na obr. 4.5. Proto je možné textově konverzovat s neomezeným počtem uživatelů.

Výsledek: Test byl úspěšně bez jakékoliv známky chybovosti. Všechny zprávy byly doručeny těm uživatelům, pro které byly určeny.



Obr. 4.5: Konverzace tří uživatelů v prohlížeči Chrome

5 ZÁVĚR

Cílem bakalářské práce bylo vytvoření funkčního SIP klienta. Tato aplikace je připojována k vlastnímu SIP serveru a umožňuje komunikaci jak textovou, tak i obrazovou se zvukem.

Prvním krokem byl průzkum technologických možností pro realizaci hovorů pomocí webového prohlížeče, kdy je hovor možné uskutečnit bez omezení spočívajících v instalaci dalšího software a klienti spolu komunikují pouze přes webovou stránku.

Pro realizaci aplikace byla vybrána knihovna JsSIP. Aby aplikace mohla být prohlášena za funkční a následně otestována, byla nutná instalace SIP serveru. V dokumentaci knihovny jsou uvedeny čtyři servery, se kterými by měla být aplikace funkční. Po opakovaném neúspěchu při realizace hovoru přes SIP servery Asterisk a Kamailio běžících na OS Linux byla provedena instalace serveru OfficeSIP v operačním systému Windows Server.

Aplikace je napsána v jazyce JavaScript. Pro některé funkce v aplikaci jsou použity jazyk PHP a technologie AJAX.

Z webových stránek projektu JsSIP byla stažena knihovna ve verzi 0.21 a byla provedena její implementace do vlastního projektu. Po vydání nové verze prohlížeče Chrome začala být knihovna nekompatibilní a aplikace nebyla použitelná.

Krátce po vydání nové verze prohlížeče Chrome byla aktualizována i knihovna JsSIP a to na verzi s označením 0.3. Tato verze knihovny byla použita pro celou aplikaci. Ovšem některé části zdrojových kódů musely být přepsány nebo nahrazeny novými kódy, neboť verze 0.21 a 0.3 nejsou zpětně kompatibilní.

Dále byla vybrána knihovna jQuery pro zjednodušení tvorby aplikace. Ve fázi ladění byla zvolena i knihovna jQuery-FancyBox, jež umožnila vytvoření jednoduchého vyskakovací okna.

V konečné fázi bylo provedeno testování funkčnosti aplikace společně se SIP serverem. Některé výsledky nebyly úspěšné. V kapitole 4 jsou kompletně popsány postupy testů a jejich výsledky.

Do budoucna bude nutné v aplikaci ošetřit výjimky. Některé výjimky jsou ošetřeny, ale současný stav není plně vyhovující. Autoři knihovny JsSIP plánují vydat v blízké době její další verzi, která by měla obsahovat metody nutné pro zjištění informací o přítomnosti uživatelů (tzv. presence information). Po vydání nové verze knihovny JsSIP by bylo užitečné tyto metody využít v aplikaci pro zobrazení aktuálního stavu uživatelů v seznamu kontaktů.

LITERATURA

- [1] SNIR, Y., et al. *Policy Quality of Service (QoS) Information Model* [online]. 2003, [cit. 24. 10. 2012]. Dostupné z URL: <<http://tools.ietf.org/html/rfc3644>>.
- [2] SPENCER, M., et al. *IAX: Inter-Asterisk eXchange Version 2* [online]. 2010, [cit. 26. 10. 2012]. Dostupné z URL: <<http://tools.ietf.org/html/rfc5456>>.
- [3] SAINT-ANDRE, P. *Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence* [online]. 2004, [cit. 26. 10. 2012]. Dostupné z URL: <<http://www.ietf.org/rfc/rfc3921.txt>>.
- [4] BERJON, R. et al. *HTML5: Edition for Web Authors* [online]. 2011, poslední aktualizace 25. 10. 2012 [cit. 11. 11. 2012]. Dostupné z URL: <<http://www.w3.org/TR/2012/WD-html5-author-20121025>>.
- [5] KOTALA, Z., TOMAN, P. *Java Applet* [online]. 2001, [cit. 13. 11. 2012]. Dostupné z URL: <<http://v1.dione.zcu.cz/java/sbornik/17.html>>.
- [6] Microsoft Corporation. *Introduction to ActiveX Controls* [online]. 2012, [cit. 13. 11. 2012]. Dostupné z URL: <[http://msdn.microsoft.com/en-us/library/aa751972\(v=vs.85\).aspx#ActiveX_controls](http://msdn.microsoft.com/en-us/library/aa751972(v=vs.85).aspx#ActiveX_controls)>.
- [7] HOLZNER, S. *JavaScript profesionálně*. Vyd. 1. Brno: Mobil Media, 2003, 1071 s. ISBN 80-865-9340-1.
- [8] The PHP Group. *PHP:PHP Manual* [online]. 2001, poslední aktualizace 24. 5. 2013 [cit. 27. 5. 2013]. Dostupné z URL: <<http://www.php.net/manual/en/intro-whatcando.php>>.
- [9] GARRETT, J. J. *Ajax: A New Approach to Web Applications* [online]. 2005, poslední aktualizace 18. 2. 2005 [cit. 22. 5. 2013]. Dostupné z URL: <<http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>>.
- [10] BERGKVIST, A., C. BRUNETT, D., JENNINGS, C., NARAYANAN, A. *Real-time Communication Between Browsers* [online]. 2011, poslední aktualizace 15. 11. 2012 [cit. 3. 12. 2012]. Dostupné z URL: <<http://dev.w3.org/2011/webRTC/editor/webRTC.html>>.
- [11] Google, Inc. *WebRTC Internals* [online]. 2011, [cit. 3. 12. 2012]. Dostupné z URL: <<http://www.webrtc.org/reference/webRTC-internals>>.

- [12] Google, Inc. *WebRTC Native APIs* [online]. 2011, [cit. 3.12.2012]. Dostupné z URL: <<http://www.webrtc.org/reference/native-apis>>.
- [13] ROSENBERG, J. et al. *SIP: Session Initiation Protocol* [online]. 2002, [cit. 22.11.2012]. Dostupné z URL: <<http://www.ietf.org/rfc/rfc3261.txt>>.
- [14] SCHULZRINNE, H., CASNER, S., FREDERICK, R., JACOBSON, V. *RTP: A Transport Protocol for Real-Time Applications* [online]. 2003, [cit. 22.11.2012]. Dostupné z URL: <<http://tools.ietf.org/html/rfc3550>>.
- [15] DONOVAN, S. *The SIP INFO Method* [online]. 2000, [cit. 24.11.2012]. Dostupné z URL: <<http://tools.ietf.org/html/rfc2976>>.
- [16] HANDLEY, M., JACOBSON, V., PERKINS, C. *SDP: Session Description Protocol* [online]. 2006, [cit. 26.11.2012]. Dostupné z URL: <<http://tools.ietf.org/html/rfc4566>>.
- [17] FETTE, I., MELNIKOV, A. *The WebSocket Protocol* [online]. 2011, [cit. 30.11.2012]. Dostupné z URL: <<http://tools.ietf.org/html/rfc6455>>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AJAX Asynchronous JavaScript and XML

API Application Programming Interface

CSS Cascading Style Sheets

GPL General Public License

HTML HyperText Markup Language

HTTP HyperText Transfer Protocol Secure

IANA Internet Assigned Numbers Authority

IAX Internet Asterisk eXchange

IETF Internet Engineering Task Force

ITU-T International Telecommunication Union - Telecommunication
Standardization Sector

JVM Java Virtual Machine

LDAP Lightweight Directory Access Protocol

MIT Massachusetts Institute of Technology

MMUSIC Multipart Multimedia Session Control

NAT Network Address Translation

PHP Hypertext Preprocessor

QoS Quality of Service

RFC Request for Comments

RTP Real-time Transport Protocol

SDP Session Description Protocol

SIP Session Initiation Protocol

SVN Subversion

TCP Transmission Control Protocol

TLS Transport Layer Security

TTL Time To Live

UDP User Datagram Protocol

URI Uniform Resource Identifier

VoIP Voice over Internet Protocol

W3C World Wide Web Consortium

WSS WebSocket Secure

XLST Extensible Stylesheet Language Transformations

XML Extensible Markup Language

XMPP Extensible Messaging and Presence Protocol

A OBSAH CD

- Elektronická verze bakalářské práce - bakalářská práce.pdf
- Složka SIP klient obsahující vlastní aplikaci
- Instalační aplikace serveru OfficeSIP 3.4 ve složce Instalační soubory na server
- Instalační aplikace jazyku PHP ve složce Instalační soubory na server
- Soubor README.doc obsahující popis instalace aplikace
- Soubor README.txt obsahující popis instalace aplikace